Libre Office: Лучше, чем OOo

Плюс: KDE 4.6 Live, XFCE 4.8, 50 приложений + еще много чего!

LINUX FORMAT

Главное в мире Linux

Debian прибыл!
Шесть месяцев проверки стабильности бесплатно с. 10

Апрель 2011 № 4 (143)

В ЭТОМ НОМЕРЕ

Советы по Linux
» Подробнейшие учебники

Обновитесь до KDE 4.6
» Плазмы больше, чем у Солнца

Gnome Shell
» Узнайте новости про дивный интерфейс к Gnome 3.0

Ищем идеальный менеджер файлов
» Мы обо всем позаботились за вас

OpenSUSE 11.4
» Что нового в популярном дистрибутиве?

Секреты сисадмина

Советы профессионалов, как сделать ваш Linux-компьютер быстрее, безопаснее и умнее

* Безопасность домашней сети
* Магия командной строки
* Запуск задач по расписанию

Плюс! Академия LPI
Отточите свои навыки, побыв в шкуре сисадмина
с. 64

Мэри Колвиг
«Мы можем сделать сотрудничество еще проще»
О продвижении проектов Mozilla с. 34

ЗАГРУЗКА
Двойной Linux
» Создайте мир гармонии на своем жестком диске

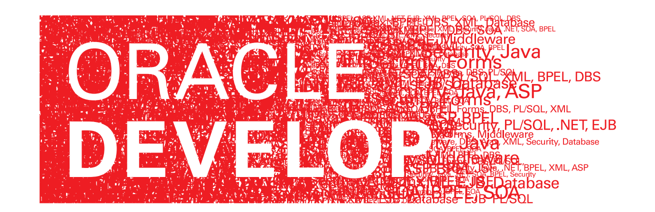
КОМАНДНАЯ СТРОКА
ExifTool
» Обработать все фото одной командой

ПРОГРАММИРОВАНИЕ
KDevelop 4.2
» Построим приложение KDE – лично свое

ПОДПИСНЫЕ ИНДЕКСЫ В КАТАЛОГАХ
Агентство «Роспечать»: годовая подписка — 36343, полугодовая подписка — 20882
«Пресса России» — 87974, «Почта России» — 16572







Впервые в Москве 2 конференции для разработчиков ПО проводятся вместе – JavaOne и Oracle Develop

12–13 апреля 2011 г., Москва

В этом году впервые эти две конференции для разработчиков ПО объединены в двухдневное мероприятие. JavaOne&OracleDevelop – это 2 дня, посвященных новейшим тенденциям в разработке программных продуктов. Последние инновации Oracle в направлении Java (Java EE, Enterprise Computing и облачные вычисления, Java SE, Client Side Technologies, and Rich User Experiences, мобильная Java), СУБД Oracle, Application Grid and Oracle WebLogic, SOA. Все это от создателей технологий и общепризнанных мировых гуру в разработке программных продуктов – мы ждем докладчиков и технологических гуру из штаб-квартиры Oracle в США.

Всю информацию найдете на сайте http://www.oracle.com/ru/javaonedevelop/index.html

Контакты:
E-mail: info@javaone2011.ru
Тел.: 8 (495) 514 25 54, 8 (495) 514 25 58

ORACLE®



Что мы делаем

» Мы поддерживаем открытое сообщество, предоставляя источник информации и площадку для обмена мнениями.

» Мы помогаем всем читателям получить от Linux максимум пользы, публикуя статьи в разделе «Учебники»: здесь каждый найдет что-то по своему вкусу.

» Мы выпускаем весь код, появляющийся на страницах раздела «Учебники», по лицензии GNU GPLv3.

» Мы стремимся предоставлять точные, актуальные и непредвзятые сведения обо всем, что касается Linux и свободного ПО.

Кто мы

Этот номер посвящен хитростям сисадминов. Мы попросили наших экспертов поделиться своими самыми глубинными Linux-секретами.



Грэм Моррисон
Однажды я написал «Linux», а «GNU/» спереди не добавил. Простите меня, пожалуйста, г-н Столлмен.



Майк Сондерс
Я стянул 20000 строк кода из UnixWare от SCO и вложил их в ядро Linux.



Нейл Ботвик
Я, как элитный пользователь Gentoo, мыслю в терминах флагов GCC. Даже велел жене **-mtune** мой кофе для i686.



Эфраин Эрнандес-Мендоса
Мигель де Икаса никогда не отвечает на мои письма. И я перестал сопровождать их поцелуями…



Эндрю Грегори
Есть только Linux между прошлым и будущим. А может быть, самая малость Hurd.



Энди Ченнел
Исповедь: я первый начал войнушку *Emacs* против *Vi*. Говорят, в нее втянулось уже 44700 человек. Каюсь…



Валентин Синицын
Признаюсь и я: это я нарочно выкладывал на каждый LXFDVD по две Ubuntu. За каждую из них мне давали селедку…



Энди Хадсон
Раз в интернет-форуме я обозвал *Wine* эмулятором. Моя почта чуть не лопнула от писем доброхотов с «поправками».



Ник Вейч
Честно-пречестно? Я не запускаю Linux на моей машине. А запускаю я его на вашей, потому что я ее взломал.



Сюзан Линтон
Секреты? Мне нечего скрывать. Даже то, что у меня есть ласкательные имена для дистрибутивов, типа Федя – Fedora. Э-э…



Шашанк Шарма
Я когда-то играл в *XBill* 47 часов подряд. Это лучший способ скоротать конференцию Microsoft.



Боб Мосс
Я знаю, что это называется Линукс, но на сборище моей LUG всегда говорю «Лайнукс». Просто чтоб быть не как все.



Весеннее оживление

«Помещая данные в компьютер, не забудь, куда ты их там положил» – так учили меня, начинающего админа, старшие товарищи. Этой шутке уже тогда было лет 20, но она актуальна и сегодня. В прошлом месяце мы сравнивали архиваторы, а сегодня поговорим о файловых менеджерах.

Весна этого года богата событиями. С истинно немецкой пунктуальностью точно в обещанный срок обновился openSUSE; соообщество LibreOffice впервые показало широкой публике результаты своей работы; после долгих месяцев ожидания наконец-то вышел Debian 6 и практически одновременно с этим, в марте, на выставке CeBIT 2011 проект Debian удостоился специальной награды за «Наиболее значительный вклад в развитие открытого и свободного ПО», что не удивительно. Об этом, а так же о новостях KDE и Gnome, смотрите в обзорах и статьях этого номера.

Мой многолетний опыт чтения самых разнообразных журналов показывает, что апрельский номер традиционно интересен в части новостей, и читать его надо особенно внимательно.

Кирилл Степанов
Главный редактор
» info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Вопросы распространения: sales@linuxformat.ru
Web-сайт: www.linuxformat.ru
› Адрес редакции: Россия, Санкт-Петербург, Лиговский пр., 50, корп. 15
› Телефон редакции: (812) 309-06-86. Дополнительная информация на с. 112



Содержание

Нам Гарри Поттер не указ. Наколдуем и без совы.



Обзоры

Debian 6.0 10
Стабильнейший из дистрибутивов наконец с нами – как никогда свободный.

› Что будем брать — Linux или FreeBSD?

KDE 4.6 11
Интересно, что-то восстанет из пылающей плазмы?

Crossover Pro 10 12
Несвященный союз приложений Windows и Linux сроду не заключался проще.

OpenSUSE 13
Проведена работа и внутри, и снаружи. Привычно-безупречно исполнен новый KDE.

› Зелененький он был... таким и остался.

Libre Office 14
OpenOffice.org, но освобожденный от контроля корпораций. Готов ли он спорить со старшими пацанами?

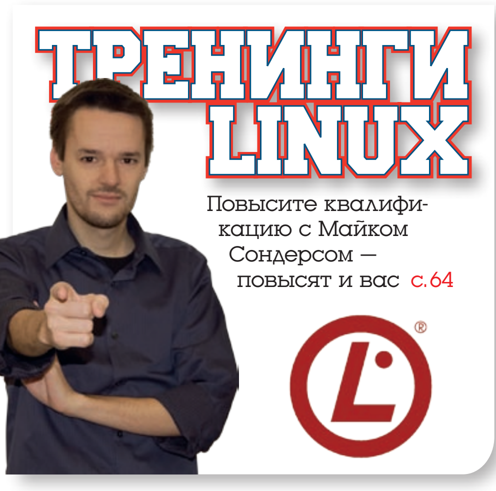
Сравнение: Файловые менеджеры

4Pane 19
Dolphin 18
Nautilus 17
PcManFM 20
Rox-Filer 20
Thunar 19



Магия сисадмина

Наводите чары с нашим руководством сисадмина и станьте всемогущим магом Linux! с. 22



Тренинги Linux

Повысите квалификацию с Майком Сондерсом — повысят и вас с. 64



Ищете работу вашей мечты? Советы от Джоно... с. 36

Люди говорят



«Люди у нас занимаются практически всем: кодом, дизайном, поддержкой...»

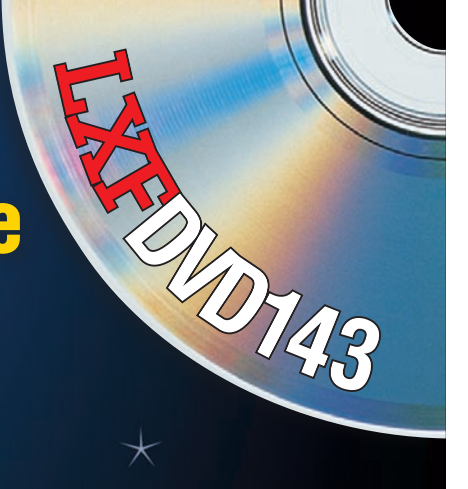
Мэри Колвиг — про волнующее сообщество с. 34

На вашем бесплатном DVD

Debian 6, KDE 4.6 и другие

» Debian: обновлен твердокаменный утес Linux
» KDE 4.6: рабочий стол немного подправили
» LibreOffice 3.3: ответвление интригует

ПЛЮС: Xfce 4.8, средства разработчика, игры, подкасты… с. 100



Ищите в этом номере...

19 перемен к лучшему 30
Критические дни для Конкретного рабочего стола: Как ему живется?

AR Drone 32
Крылья у Linux были всегда, а теперь он выучился летать. Взгляните на аэро-ОС!

Загрузка на марше 46
Док оценивает нового претендента на революционизацию процесса загрузки: *systemd*.

Это модное слово «виджет» 72
Если красота – тоже функция, почему бы не украсить свою систему.

Пропустили номер?

Узнайте на с. 107, как получить его прямо сейчас!



Постоянные рубрики

Новости 4
Debian получил награду CeBIT, а свободу будут носить в кармане.

Сравнение 16
Доверьте управление своими файлами этим превосходным топ-менеджерам.

Интервью LXF 34
Мэри Колвиг добежала до Mozilla, чтобы вовлекать народ в *Firefox*.

Что за штука 44
Gnome Shell крадется в планшеты и мобильные устройства, прячась в тени Unity.

Рубрика сисадмина 46
Добреремся до ядра истины и удаленных серверов и изучим загрузку с *systemd*.

Ответы 88
ПРОБЛЕМЫ LINUX РЕШЕНЫ!
Говорите, индикаторы мигают как-то не так? Мы поможем.

Hotpicks 94
Отведайте горяченького: лучшие в мире новинки свободного ПО.

Диск *Linux Format* 105
Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 107
Еще не поздно заказать любой из предыдущих выпусков журнала.

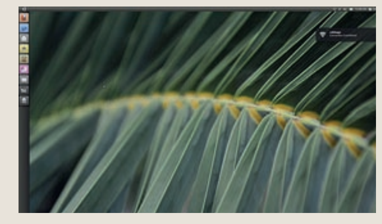
Школа LXF 108
Кто заявлял, что Linux – препятствие к сдаче ЕГЭ, сказал чушь. Эта ОС уважаема даже международными олимпиадами.

Через месяц 112
Тащите свой магический кристалл, и вместе поглазеем на чудеса следующего номера.

Учебники



Начинающим
Gnome 52
Добро пожаловать в мир чудес Linux! Надеемся, вам здесь понравится. Садитесь и послушайте про дружелюбный рабочий стол.



› Угадайте: это Gnome Shell или Unity?

Двойная загрузка
Linux и… Linux. Два Linux'а ... 56
Поможем Fedora и Ubuntu достичь гармонии в совместной жизни.

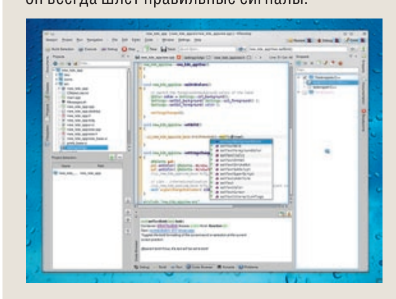
Управление проектами
Todoyu 60
Установив этот мощный инструмент, вы сделаете свою работу небывало эффективной.

Образование
Тренинги Linux 64
Школьные дни чудесные – спасибо Майку: с ним не страшен экзамен на сертификат. В этом месяце: файловые системы.

Программирование
ExifTool 68
Порядок в вашей коллекции фото обеспечен – инструмент обработки EXIF-данных и наши скрипты мигом его наведут.

Программирование
QML, Mono и Erlang 72
Три Андрея-полиглота излагают начала и концы умных языков программирования.

Hardcore
KDevelop 84
Делаем текстовый редактор с Грэмом: он всегда шлет правильные сигналы.



› Не робейте, пробуйте Kdevelop!

Новости

ГЛАВНОЕ Debian удостоен награды » Карманные серверы » Открытым бывает не только ПО, но и «железо» » Новый вид моделизма? » Коротко о разном

СВЕТСКАЯ ЖИЗНЬ

Награды CeBIT 2011



» Рубрику готовил **ТИМУР МУБАРАКШИН**

На выставке в Ганновере оценивали свежие достижения в области открытых технологий. Героем дня оказался почтенный Debian.

На прошедшей в начале марта в Ганновере (Германия) выставке CeBIT 2011 состоялась церемония награждения лауреатов конкурса Linux New Media Awards. В рамках этой престижной премии были награждены семь лучших открытых проектов, связанных с Linux. Стоит отметить, что в этом году впервые использовалась новая методика оценки проектов при выявлении лучшего дистрибутива Linux – общественное голосование, организованное в Сети. В прочих категориях победители были отобраны жюри, в которое вошли 200 наиболее известных представителей сообщества разработчиков свободного ПО, связанных с Linux изданий и экспертов из компаний, применяющих открытую модель разработки. Заметим, что почтенный дистрибутив Debian/GNU Linux одержал победу в двух номинациях – «Лучший серверный дистрибутив» и «Наиболее значительный вклад в развитие открытого и свободного ПО». В 2010 году первое место заняли разработчики Linux-ядра, а старина Debian удостоился почётного второго места.

Как видите, в этом году заслуги команды Debian были оценены очень высоко. При вручении дедушке многих дистрибутивов первой премии были отмечены не только инновационные технологии, реализуемые в Debian, но также и достижения в формировании стандартов и процессов разработки свободного ПО. По словам Карстена Герлоффа [Karsten Gerloff], при разработке Debian основное внимание уделяется обеспечению высокого качества дистрибутива. Но разработчики также понимают, что самая замечательная программа ничего не стоит без свободы. В последнем стабильном релизе Squeeze воплощён в жизнь план по обеспечению поставки полностью свободного ядра, не содержащего кода проприетарных двоичных прошивок. Debian стал первым из основных Linux-дистрибутивов, который решился на подобный шаг.

> «Почтенный дистрибутив одержал победу в двух номинациях.»

Также на CeBIT 2011 были награждены: высокопроизводительная файловая система для серверов и рабочих станций btrfs (в номинации «Самый инновационный открытый/свободный проект»); интернет-планшет Samsung Tablet Galaxy под управлением вездесущей мобильной ОС Android (в номинации «Наиболее интересное Linux-устройство»); мобильная версия самого популярного в Европе браузера *Firefox* – *Fennec* (в номинации «Лучшее приложение для мобильных устройств»); система виртуализации уровня ядра *KVM* (в номинации «Лучшее открытое решение в области облачных технологий и виртуализации»). А лучшим дистрибутивом 2011 года признан Ubuntu – именно его предпочло большинство при общественном голосовании.

Приятно, что год от года всё больше открытых проектов удостаиваются престижных премий.



› **Групповое фото: победители и ведущие Linux New Media Awards 2011.**

› **Александр Райхель-Шмель [Alexander Reichel-Schmehl], активный разработчик Debian, принимает заслуженную награду.**

Про CeBIT

CeBIT – это выставка, ежегодно проходящая в Ганновере, что в Германии. CeBIT (от нем. Centrum der Büro- und Informationstechnik– центр офисных и информационных технологий) – крупнейшая в мире международная выставка, посвященная информационным и телекоммуникационным технологиям. Проводится каждую весну компанией Deutsche Messe AG в Ганновере и является важным индикатором сферы информационных технологий.

ПРАВО НА ПРИВАТНОСТЬ

FreedomBox: коробочка свободы

Концепция FreedomBox призвана доставить пользователям Всемирной сети истинную свободу, безопасность и анонимность.

Когда-то, на заре Всемирной паутины считалось, что анонимность в Сети – это нечто непоколебимое и само собой разумеющееся. В начале и середине 1990-х годов не было ни Facebook, ни Вконтакте, и Анонимус, ныне снова набирающий популярность, был защищён. Шутки шутками, а в современном мире конфиденциальность частной информации сильно пострадала. Достаточно вспомнить громкие скандалы со взломами пользовательской базы Вконтакте в 2009 или серверов компании Google, а также историю с WikiLeaks. Да, из-за облачных технологий наша с вами информационная безопасность под угрозой. Однако есть альтернатива облачным системам и большим централизованным базам данных – это FreedomBox. Что такое FreedomBox? Это гарантия сохранности ваших данных, при желании – полная анонимность, и это – открытая технология, работающая на Linux. Если коротко, то FreedomBox – это концепция карманного сервера для обеспечения безопасности и приватности в Сети.

Автор FreedomBox – Эбен Моглен [Eben Moglen], профессор права Колумбийского университета, занимающийся юридической поддержкой Фонда свободного ПО. Согласно этой концепции, планируется развить идею создания миниатюрного сервера, который позволит освободить отдельных людей от «тирании клиент-серверной модели» современных web-сервисов. FreedomBox позволит пользователю вернуть контроль над своими персональными данными, обеспечить защиту их конфиденциальности и предотвратить утечки информации.

Аппаратно FreedomBox будет представлять собой дешёвый сервер с низким энергопотреблением в виде небольшого штекера, работающий под управлением Debian GNU/Linux. Маленькие «штекерные» серверы Pogoplug и TonidoPlug уже есть в продаже по цене 99 долларов. По словам Моглена, «они очень дёшевы и очень быстры... Когда все станут покупать их, они будут продаваться уже по 29 долларов».

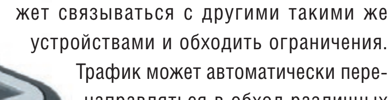
В данный момент Моглен сфокусировал своё внимание на ПО для подобных устройств, так как аппаратная составляющая в том или ином виде уже существует, и необходимы хорошие программные компоненты, чтобы сделать FreedomBox по-настоящему работоспособным продуктом. Разработчики уверяют, что устройство FreedomBox сравнится с компьютером под управлением Debian GNU/Linux, что означает доступ к тысячам пакетов приложений и функции автоматического обновления.

На базе устройств FreedomBox планируется сформировать распределённую пиринговую сеть, автоматически и надёжно предоставляющую множество сервисов, таких как:

» Безопасные социальные сети, в которых пользователь, не теряя связей ни с кем из своих друзей, перейдёт от прямого использования таких популярных централизованных систем, как Facebook, Flickr и Twitter, к обеспечивающим приватность распределённым сервисам.

» Безопасное резервное копирование. Данные пользователя автоматически сохраняются в зашифрованном виде на устройствах его друзей и коллег, работающих под управлением платформы FreedomBox, защищая таким образом персональные данные от утечки или кражи.

» Защита сетевого нейтралитета. Если провайдер начинает ограничивать доступ пользователя в Интернет или вклиниваться в трафик, устройство FreedomBox сможет связываться с другими такими же устройствами и обходить ограничения. Трафик может автоматически перенаправляться в обход различных репрессивных национальных межсетевых экранов.





› Вот такие коробочки скоро могут появиться в каждом доме и офисе.

» Безопасная анонимная публикация информации. Друзья или партнёры вне зоны сетевой цензуры смогут автоматически передавать/принимать информацию от людей, находящихся за межсетевым экраном, что позволит обеспечить людям в странах с цензурой безопасную и анонимную публикацию информации.

» Безопасность домашней сети, с реально работающей системой защиты от вторжений и угроз безопасности, связанных с атаками на Windows или на другие незащищённые места в сети.

» Электронная почта с бесшовным шифрованием/дешифрованием.

» Защищённая голосовая связь. Пользователи FreedomBox смогут совершать звонки посредством технологии VoIP друг другу или на любой другой телефон. Звонки между пользователями FreedomBox будут надёжно зашифрованы.

Для координации усилий в области развития проекта FreedomBox Моглен создал некоммерческую организацию Freedom Box Foundation. По словам Моглена, «если получится добыть чуть меньше 500 тысяч долларов, FreedomBox 1.0 будет готов в течение одного года».

Кстати, ещё в октябре 2010 года министр связи и массовых коммуникаций Российской Федерации Игорь Щёголев заявил, что Международный союз электросвязи способен управлять Интернетом, а значит, управлять Интернетом всё-таки можно. Подобные заявления официальных лиц не могут не вызывать опасений, ибо в современном бушующем Мире нет никакой гарантии, что «управляющие» не будут использовать свою силу в чьих-то корыстных интересах.

На фоне развития скандала, связанного с WikiLeaks и заявлений официальных лиц различных государств о необходимости контроля над Интернетом, FreedomBox смотрится как единственно работающая альтернатива тотальному контролю. И весьма отрадно, что данная технология работает на Linux.

Свобода помещается в кармане

«Карманные серверы» – Pocket Servers появились сравнительно недавно, и пока не получили широкого распространения; но уже есть работающие и продающиеся образцы этих малышей – два родственных устройства с похожими названиями: TonigoPlug и PopoPlug являются яркими представителями семейства «карманных серверов». Несмотря на свои маленькие размеры, они обеспечивают многие функции, ранее доступные только на их «больших братьях». Всё это стало возможным благодаря уникальной масштабируемости и открытости Linux. Также бытует мнение, что в первое время децентрализованные сети из таких карманных серверов будут немного отставать по скорости работы от таких крупных централизованных сервисов, как Amazon EC2, но ведь и BitTorrent в начале был не таким уж и шустрым. А сейчас это один из основных протоколов обмена файлами в Сети.

НЕ ТОЛЬКО ПРОГРАММЫ

Open Source Hardware – быть!

Открытое аппаратное обеспечение обзавелось еще одним документом, стандартизирующим и унифицирующим разработку.

Знаете ли вы, что кроме открытого программного обеспечения есть ещё и открытое «железо»? Так вот, OSHW – это компьютерное и электронное аппаратное обеспечение, разработанное в том же стиле, что и свободное и открытое программное обеспечение. Это часть открытой культуры, которая относит идеи открытого доступа и к иным областям применения (не только ПО). В прошлом номере нашего журнала мы уже упоминали о принятии определения открытого аппаратного обеспечения (Open Hardware Definition 1.0).

Рассмотрим эту тему подробнее. Совсем недавно, в первой половине февраля сего года, был представлен финальный вариант документа “Open Source Hardware Definition 1.0 (OSHW)”, в котором определён список основных требований, необходимых и достаточных для признания открытой лицензии на аппаратный проект. Документ OSHW является адаптированным для аппаратного обеспечения аналогом правил Open Source Definition, определяющих условия открытости лицензий на программное обеспечение.

OSHW определяет 12 базовых требований, которые нужно соблюдать при создании открытого аппаратного обеспечения. Ключевые требования:

» Обязательность поставки полной документации и программного обеспечения (прошивки), допуская их модификацию. Обязательное свободное распространение схем.

» В документации должно быть явно указано, какие части дизайна выпущены под открытой лицензией.

» Возможность модификации и создания производных продуктов, распространяемых на условиях оригинальной разработки. Лицензия должна давать возможность производить, продавать, распространять и использовать продукты, созданные по представленным спецификациям.

» Лицензия не должна ограничивать право на продажу или заимствование проектной документации, не должна требовать отчислений с продаж.

» Соблюдение целостности оригинального кода автора, сторонние изменения должны быть выделены явным образом – например, сменой названия или другим номером версии.

> «Несомненно, концепция OSHW очень интересна и перспективна.»

» Преемственность лицензии. Переданные лицензией права должны распространяться на все дополнения к разработке; производные продукты не должны подвергаться дополнительному лицензированию.

» Отсутствие привязки лицензии к заданному продукту.

» Недопустимость дискриминации по области применения или использованию определёнными группами лиц.



› **Tuxphone — выглядит страшновато, но зато это свободное аппратное обеспечение!**

» Лицензия не должна ущемлять свободу других аппаратных или программных систем.

» Лицензия должна быть технологически нейтральной и не должна навязывать выбор тех или иных технологий, материалов или оформления интерфейса.

Документ подписали представители многих организаций, так или иначе связанных с миром открытых технологий, такие как, например, Дэвид Мэллис [David A. Mellis]; Дэвид Картеллис [David Cuartielles] и Массимо Банци [Massimo Banzi] из проекта Arduino, в рамках которого разрабатывается открытое аппаратное обеспечение; Крис Уокер [Chris Walker] из проекта Netduino – открытой платформы электрических схем, работающей на базе открытого фреймворка .NET Micro Framework; Дэвид Кэрриер [David Carrier] из компании Parallax Inc.; Джим Баркли [Jim Barkley] и Самюэль Сойер [Samuel Sawyer] из научной организации MITRE Corporation; и даже один представитель сообщества OpenSuse – Федерико Лучифреди [Federico Lucifredi], а также многие другие, кому небезразлична эта тема. С подробным списком подписавшихся под новым документом можно ознакомиться на wiki-страничке проекта.

Несомненно, концепция OSHW очень интересна и перспективна, однако предстоит ещё большая работа по разработке стандартов оборудования и принципов построения экосистемы открытого аппаратного обеспечения. Так что любителям и профессионалам есть где развернуться!

Открытость до Луны довезёт

В мире открытого аппаратного обеспечения есть интересные разработки: например, ставшая уже популярной среди радиолюбителей Arduino – аппаратная вычислительная платформа, основными компонентами которой являются простая плата ввода/вывода и среда разработки, или проект RepRap – инициатива, направленная на создание самокопирующегося устройства, которое может быть использовано для быстрого прототипирования и производства: устройство RepRap представляет собой 3D-принтер, способный создавать объёмные детали на основе моделей, сгенерированных компьютером; или же OpenSPARC– проект по разработке микропроцессоров архитектуры SPARC с открытым кодом, начатый в своё время знаменитой Sun; для любителей открытых мобильных устройств есть даже целый сотовый телефон – TuxPhone, разрабатываемый энтузиастами; есть даже совсем уже космический в прямом смысле этого слова проект – Lunar Numbat, разрабатываемый в рамках конкурса Google Lunar X-prize: целью проекта является посадка на Луну космического беспилотного аппарата-робота под управлением Linux. Группа австралийских Linux-энтузиастов использует свободное программное и аппаратное обеспечение для проектирования космического аппарата, который сможет прилуниться и выиграть приз от Google – 30 миллионов долларов США. Дерзайте, и кто знает – может, именно ваш аппарат под управлением Linux станет первым космическим кораблём, привезшим человека на Марс!

НАВСТРЕЧУ СЛАВНОМУ ЮБИЛЕЮ

ВЦ АН СССР на книжной полке

В нашей стране родился новый вид моделизма – моделирование старых компьютеров. О его перспективах рассказывают **Кирилл Степанов** и **Алексей Федорчук**.

Вероятно, когда вы будете читать эти строки, уже наступит 12 апреля, и весь мир будет отмечать 50-летие первого полета человека в космос. Знаменитую королевскую «Семерку» – ракету-носитель Р-7 – можно своими глазами увидеть в Москве или Самаре, и ее вид хорошо знаком всем, кто интересуется историей космонавтики. А вот людей, кто помнит, как выглядели некоторые другие машины, сыгравшие не менее важную роль в успехе полета Гагарина, становится все меньше и меньше... Речь идет об отечественных компьютерах, или как их именовали в ту пору, ЭВМ, выпуска 50-60-х годов, на которых и выполнялись сложнейшие баллистические и инженерные расчеты. К сожалению, некогда знаменитые БЭСМы и «Уралы» невозможно поставить на постамент, как легендарные Т-34 или Р-7, а ведь они не в меньшей степени того достойны...

Но не все так плохо. По случаю 50-летия выхода человека в космос и для сохранения памяти о годах, когда престиж отечественной техники и технологии во всем мире был высок как никогда, группа российских предприятий и организаций, а также энтузиастов-историков ИТ начинают выпуск серии моделей отечественных ЭВМ в масштабе 1:43. Это не опечатка – выбран «классический» масштаб для моделей автомобилей. Но и это не главное. Все модели – действующие! В отличие от уже привычных виртуальных машин, они выполнены во вполне реальных «железе» и кремнии. Их можно подержать в руках и поставить на полку – выбранный масштаб для этого очень удобен.

Пока все модели строятся базе отечественного микропроцессора 1В579 (набор команд SPARC), на котором выполняется эмулятор соответствующего «старого железа». Модель подключается к компьютеру по USB, что позволяет запускать на ней самые настоящие программы от ЭВМ-оригинала. Для этого понадобится установить соответствующий эмулятор терминала и устройств ввода-вывода (магнитных барабанов, считывателей перфолент и перфокарт). Дальнейшие работы по масштабированию элементной базы пройдут в несколько этапов – сначала будет освоена техника эмуляции лампочек для пультов, а потом и до ламп для процессора дело дойдёт. Специалисты «РОСНАНО», генерального спонсора всего проекта, напряжённо работают над этой проблемой уже сегодня.

Авторы обещают, что в самом начале апреля исходные коды ПО и схемы аппаратуры будут опубликованы под свободной лицензией – ЗСиЧП. То есть под лицензией Здравого Смысла и Человеческой Порядочности. Той самой, под которой испокон века распространялись достижения отечественной науки. Да пожалуй что и мировой тоже.

На момент написания этой заметки к выпуску готовились модели ЭВМ БЭСМ-6 и «Урал-1». В планах же – «Днепр», «Мир-1», «Минск-32», М-20, М-40 и 5Э26, а также бортовые ЭВМ серий «Пламя» и «Аргон», хотя последние, возможно, будут выполнены в другом масштабе.

> «На ЭВМ выполнялись сложнейшие баллистические расчеты.»

Работа представляет собой не только исторический интерес. Предполагается, что путём моделирования лучших достижений отечественного компьютеростроения удастся эмулировать и творческую мысль их создателей – и тем самым способствовать возрождению отечественных высоких технологий. Так что будем надеяться на успех и внимательно следить за новостями. LXF



Электронная вычислительная машина «Урал-1».

Фото © www.wikipedia.org

Новости короткой строкой

- Вышла третья альфа-версия Ubuntu 11.04 “Natty Narwhal”. 28 апреля ожидается финальный релиз.
- Компания Google выпустила очередную тестовую сборку программной платформы.
- В Китае появится новая мобильная ОС WoPhone на базе ядра Linux, от компании China United Network Communications Group: для смартфонов и планшетных компьютеров.
- Разработчики Mozilla представили первый тестовый релиз проекта WebApps, развивающего инфраструктуру для разработки и распространения локальных web-приложений.
- Злоумышленники блокировали закрытую рассылку с обсуждением уязвимостей инфраструктуры lst.de, наглядно показав ущербность практики скрытого устранения уязвимостей.
- Анонсирован релиз дистрибутива Scientific Linux 6.0 на пакетной базе Red Hat Enterprise Linux 6.0, для научных учреждений.
- Microsoft объявила о переводе на открытую платформу WordPress более миллиона блогов из сервиса Windows Live. Оставшимся пользователям предложат совершить миграцию.
- Google удалила из каталога Android Market 56 приложений, инфицированных вирусом DroidDream. Заразилось от 50 до 200 тысяч телефонов.
- Компания «Пингвин Софтвер» стала первой российской организацией в Linux Foundation.
- Разработка финальной версии *Firefox 4* вышла на финишную прямую. Релиз намечен на 22 марта 2011 г.
- Проект открытого медиа-центра *Boxee* получил $16,5 млн от компаний Pitango и Softbank, на расширение TV-сервисов и налаживание партнерских отношений по выпуску телеприставок.

LINUX FORMAT Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов



АЛЕКСЕЙ ФЕДОРЧУК
...простой советский юзер.

ОС Barrelfish: рыбозасолочный цех

Разработчики не часто удивляют нас появлением новых операционных систем. Оно и понятно: казалось бы, в существующих ОС реализованы все разумные идеи. Ан нет: осенью 2009 года мы имели удовольствие видеть представление Barrelfish – ОС с принципиально новой, мультиядерной [multikernel] архитектурой: в ней, подобно сельдям в бочке, несколько ядер [kernel], соответствующих ядрам [core] аппаратной платформы, работает независимо, с собственными приложениями.

Возможно, такой «рыбозасолочный» подход будет более эффективным при мультипроцессинге, нежели традиционные механизмы SMP или NUMA. А ведь именно эффективное распараллеливание задач между наличными ядрами – ныне единственный способ наращивания производительности, особенно ввиду скорого появления восьмиядерных процессоров даже в пользовательских десктопах.

Интересны разработчики и лицензия новой ОС. Первые – Высшая техническая школа Цюриха (ETHZ), известная многими именами учёных в области точных и компьютерных наук, и... компания Microsoft, просто известная. А лицензия – практически стандартная в BSD-стиле.

Первого марта текущего года появился третий по счёту снапшот (подробности на http://www.barrelfish.org/). Система позиционируется как исследовательская, и к практическому применению (пока) не пригодна. Однако – кто знает? Может быть, скоро мы увидим действующий «рыбозасолочный цех», порождённый альянсом научной свободы и проприетаризма.

alv@posix.ru

Сегодня мы рассматриваем:

Debian 6 10
Только фильмы по трилогии «Властелин колец» потребовали больше времени на их выпуск. Но, в отличие от киношек, Debian – особая статья для работы в Linux. А теперь и в BSD.

KDE 4.6 11
KDE наконец-то добился релизов, где новинок все же больше, чем поломок. Версия 4.6 улучшила контроль приложений над Действиями и ввела новые функции в *Dolphin*. Авось ей удастся смягчить горечь утраты для фанатов серии 3.x.

CrossOver Pro 10 12
Знаковый релиз всеми любимого «слоя совместимости с Windows». В нем тонны усовершенствований и куча готовых капсул для популярных программ – вы почти наверняка найдете интересующее вас приложение.

OpenSUSE 11.4 13
Назло анонимным интернет-аналитикам, openSUSE гораздо более жив, чем мертв, и чувствует себя отлично: например, первым из главных дистрибутивов рискнул обзавестись *LibreOffice*.

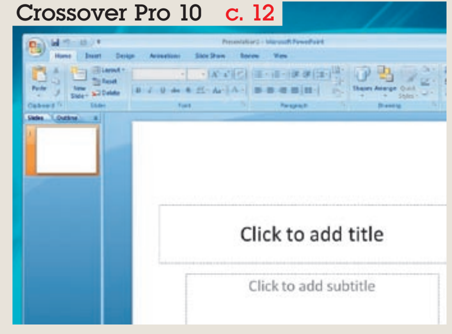
LibreOffice 3.3 14
Прибыл один из самых предвкушаемых выпусков свободного ПО за последние шесть месяцев. Этак получится, что *OpenOffice.org* угодит в мусорную корзину – вдобавок при установке новичок ведет себя как кукушонок, выпихивая старшего брата из гнезда в системе.

Debian 6 с. 10



› Не прошло и трех лет разработки, как появился Debian 6. Его очередь ждать вас — на нашем DVD!

Crossover Pro 10 с. 12



› Решение для тех, кто не может обойтись без *MS Office* и *Photoshop* от Adobe.

Наш вердикт: Пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибалльной шкале (0 – низшая оценка, 10 – высшая). Как правило, учитываются функциональность, производительность, простота использования и цена, а для бесплатных программ – еще и документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.

LINUX FORMAT TOP STUFF

Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но если разработчики рекомендуют *Autopackage*, мы следуем этому совету.

LINUX FORMAT Вердикт

Google Earth

Разработчики: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Оправданность цены	9/10

» Если весь мир — сцена, то Google Earth — театр. Простая в использовании, захватывающая и ободряюще практичная программа.

Рейтинг 9/10

Debian 6.0



Боба Мосса обуревают сомнения: не обернется ли слишком упертая «свободность» – стерильностью?

Вкратце

» **Дистрибутив, приверженный принципам FSF в отношении свободы ПО. Благодаря своей стабильности, часто служит основой для производных дистрибутивов. См. также:** *Gentoo, Fedora, Ubuntu.*

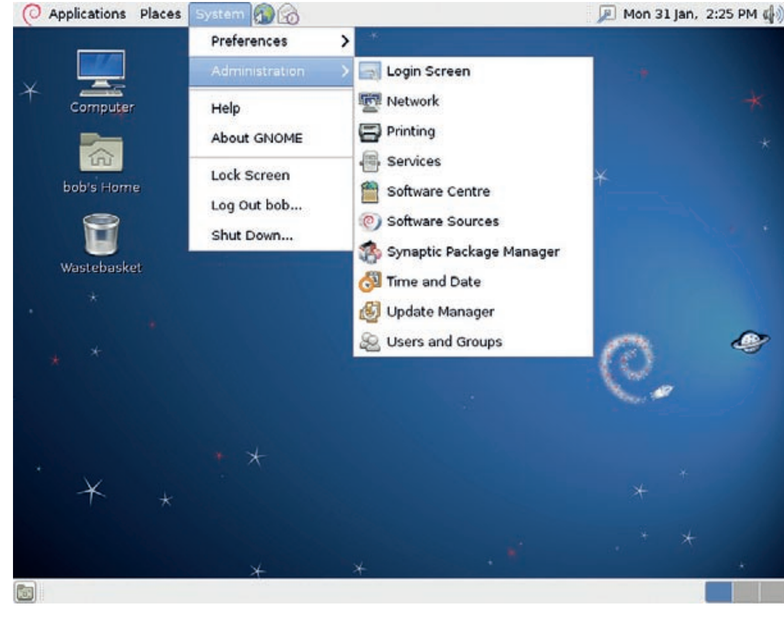
Ключевое ПО

» Gnome 2.3
» KDE 4.4.5
» LXDE 0.5
» XFCE 4.6.2
» X.org 7.5
» OpenOffice 3.2
» GCC 4.4
» PHP 5.3.2
» Apache 2.2
» MySQL 5.1

Версию Debian 6.0 (Squeeze) ждали долго. Если вы не в курсе событий, напомним: первоначально стабилизация была назначена на март 2010 г., но впоследствии отложена на полгода из-за многочисленных недостатков. Затем мы узнали, что новая версия выйдет в конце 2010 г., и были весьма разочарованы, увидев к тому времени лишь релиз-кандидат. В феврале 2011 г. Squeeze-таки увидел свет, и теперь мы можем ответить на вопрос, стоило ли ждать долгие шесть месяцев.

Сразу оговоримся, Debian всегда был монументально стабилен – не стал исключением и Squeeze. Но стабильность достигается за счёт отказа от восхитительных новинок, которыми пестрят другие дистрибутивы. Взгляните на список ключевого ПО (слева): у большинства пакетов весьма почтенный возраст. Поэтому-то Debian и не блещет новыми функциями. Новая графика радует, но без нее можно и обойтись. Замечена версия *Ubuntu Software Centre* без бренда, пара новых клавиатурных раскладок для второстепенных языков да несколько исправленных ошибок для финского.

Стоит упомянуть о том, что проект Debian разрабатывается под эгидой Фонда свободного ПО (FSF), поэтому разработчики стремятся сделать систему максимально свободной. Squeeze продолжил эту тенденцию, исключив несвободные компоненты из ядра. Однако применение *Epiphany* как браузера по умолчанию (при наличии *Iceweasel*) остается для нас загадкой. Безусловно, *Iceweasel* (тот же *Firefox*, только под другим названием) обеспечивает более развитую функциональность и работает с большинством сайтов лучше, чем *Epiphany*.



» **Как будто Debian и без того недостаточно свободен: взяли и убрали «лишние» компоненты из ядра Linux. Р-р-р…**

Впрочем, одна новинка выламывается из монолитной нерушимости Debian: это официальный порт BSD. Несмотря на то, что ПО не лишено ошибок, приятно видеть попытку солидного дистрибутива выйти за рамки Linux-ядра.

Идти налево?

Debian часто используют на автоматических устройствах – например, роутерах; но как он покажет себя в качестве рабочего дистрибутива? Здесь всё не так уж безоблачно. Тестируя дистрибутив, мы обнаружили, что отсутствие проприетарных драйверов ослабляет поддержку аппаратного обеспечения. А если вы привыкли слушать файлы MP3 или смотреть DVD-фильмы, придётся добавить собственные репозитории и загрузить несвободные кодеки вручную (подсказок не ждите: их нет). Версии ПО, лишённые брендов, зачастую лишаются и новых функций или теряют совместимость с некоторыми модулями (*Iceweasel* тому яркий пример).

Однако в таких сферах, как загрузчик *Grub* или загрузочные экраны, Debian держит марку; и мы ни разу не столкнулись с зависанием дистрибутива или поддерживаемого приложения при нормальной работе.

Словом, для стоика, презирающего греховную несвободу и обладающего терпением святого, лишние шесть месяцев ожидания Squeeze вполне оправданы – стабильность новой версии, как всегда, на высоте. Но если вам дорога коллекция медиа-файлов и вы следите за прогрессом в мире ПО, поищите другой дистрибутив. LXF



Свойства навскидку

Software Centre
Знакомый интерфейс? В Squeeze включили «обезличенную» версию Ubuntu Software Centre.

Инсталлятор Cooler
Кроме того, Squeeze получил обновлённый, гораздо более удобный инсталлятор для загрузки программ.



Вердикт

Debian 6.0

Разработчик: Debian
Сайт: www.debian.org
Цена: Бесплатно на условиях GPL

Функциональность	6/10
Быстродействие	9/10
Удобство использования	6/10
Документация	6/10

» Самый стабильный из существующих дистрибутивов, но вас ждет испытание на преданность свободному ПО.

Рейтинг 7/10

KDE 4.6



Вот и седьмой выпуск KDE 4. Продолжается ли погоня за инновациями, или вперед вышли вопросы удобства? **Нейл Ботвик** погрузился в *Plasma*…

Вкратце

» Комплексная среда рабочего стола с богатым набором новейших функций и дизайнерских находок. См. также *Gnome*, XFCE.

Как и всегда в начале года, мы задаёмся вопросом: «Станет ли этот год годом Linux на рабочем столе?» Теперь к этому добавился ещё один вопрос: «Станет ли этот год годом KDE на рабочем столе Linux?» Итак, вышла очередная версия KDE – 4.6.

Первые варианты KDE 4 разочаровывали, но пара последних версий (особенно 4.5) стали получше. А как насчет 4.6? Удалось ли продвинуться по сравнению с предыдущей версией, и сможет ли KDE по-прежнему достойно конкурировать с другими рабочими столами?

Activities

Одно из самых заметных новшеств этой версии – сближение интерфейса Plasma Desktop с рабочими столами для нетбуков. Система Activities, ранее вносившая изрядную путаницу, стала гораздо логичнее.

В меню, вызываемому щелчком правой кнопки мыши на строке заголовка окна, введен пункт Activities, что упрощает добавление окон в эту систему. Управление системой Activities стало интуитивнее. Для каждого экземпляра можно назначить свой комплект виджетов и обоев. Это не виртуальные рабочие столы, поскольку Activities накрывает все рабочие столы; скорее это размноженные «рабочие столы на стероидах». Но если вам ближе простой подход, используйте отдельный экземпляр Activity для каждого рабочего стола.



› Виджеты здесь повсюду, но и для полезной работы места достаточно. А всё лишнее можно разом скрыть.

> «Стабильность была и до этого, но в KDE 4.6 удобнее работать.»

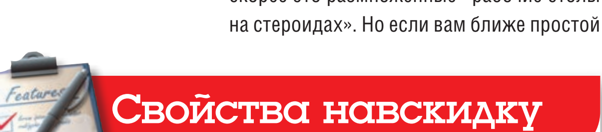
Файловый менеджер *Dolphin* научился работать в режиме «многогранного обзора», облегчающем поиск файлов. Можно искать по имени, фильтровать по типам файлов, ограничивать совпадение датами – или комбинировать все эти критерии. Возможен и поиск по содержимому (в т.ч. по метаданным EXIF для фото), поскольку KDE постоянно переиндексирует файлы в вашем каталоге **home**. Индексацию можно отключить – например, на маломощной машине. А на ноутбуке можно задать ее выполнение только при подключении к питанию, с помощью функции расширенного управления питанием.

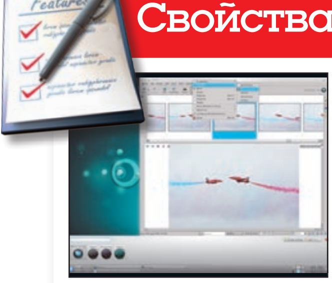
Приложения типа *Gwenview* и *KSnapshot* теперь умеют отсылать фотографии прямо на сайты с галереями или в социальные сети. Увы, в KDE всё ещё нет средств работы с сетью. Лишив владельцев современных ноутбуков (с расширенными настройками питания) доступа к беспроводной сети, популярности у них не завоюешь.

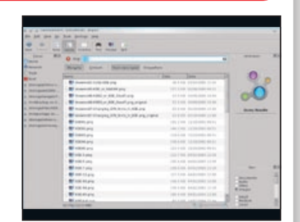
Интерфейс нетбука

Свое понятие об идеале интерфейса для нетбука есть у каждого – не избежал поветрия и KDE. Что ж, версия нетбука с *Plasma* имеет право на жизнь: все прочие варианты применяют *GTK*, а не «родной» формат KDE *Qt*, и это был бы единственный интерфейс пользователя, сочетающийся с KDE-приложениями. Но это не главный его плюс: он грамотно использует ограниченную экранную площадь. Все приложения работают в полноэкранном режиме, но есть удобный доступ к программам и файлам.

Все версии KDE 4 претендовали на совершенство. 4.4 была неплохой; 4.5 стала ещё лучше; ну, а 4.6 – очередной шаг в верном направлении. Стабильность была и до этого, но в версии KDE 4.6 удобнее работать. Те, кто до сих пор не пережил горечь потери KDE 3.5, будут ей рады. **LXF**

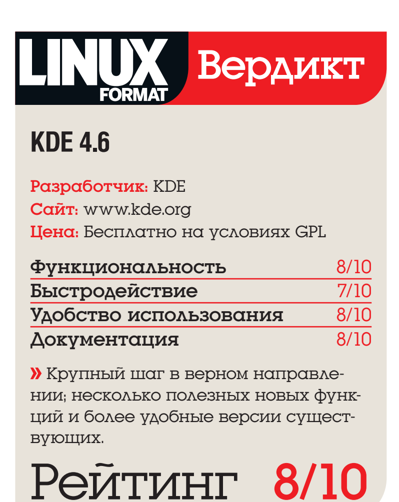






Activities
Система Activities – уже не новость; вновь будет удобство работы с ней.

«Многогранный» обзор
Dolphin упрощает работу с результатами поиска файлов.



LINUX FORMAT Вердикт

KDE 4.6

Разработчик: KDE
Сайт: www.kde.org
Цена: Бесплатно на условиях GPL

Функциональность	8/10
Быстродействие	7/10
Удобство использования	8/10
Документация	8/10

» Крупный шаг в верном направлении; несколько полезных новых функций и более удобные версии существующих.

Рейтинг 8/10

CrossOver Pro 10

В новой версии продукта Codeweavers установка упрямых Windows-приложений проще простого. **Энди Ченнел** радуется как дитя…

Вкратце

» Проприетарный комплект совместимости, позволяющий запускать приложения Windows на Linux-машине. См. также: *Parallels*, *VirtualBox*, *Wine*.

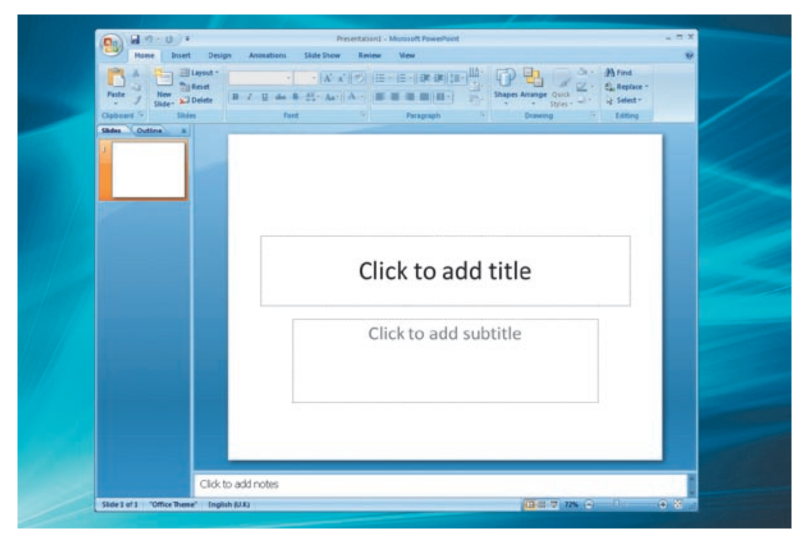
При всём величии свободного ПО бывают случаи, когда ни одно Linux-приложение не может открыть присланный вам *Word*-файл. Здесь-то и поможет пакет Windows-совместимости (*CrossOver 10*, например): не покидая комфорта своего дистрибутива, вы можете запускать любые Windows-программы.

CrossOver 10 поставляется в нескольких вариантах, предназначенных либо для игр, либо для работы. Конечно, в игровой версии *CrossOver* запустится и рабочее ПО, как и наоборот, но разница в оптимизации может быть весьма существенной. Кстати, беглое сравнение показало, что разница в быстродействии *Office 2007* под Windows и в *CrossOver Pro* совсем невелика. *Powerpoint* выполняет некоторые ресурсоёмкие операции под Linux даже чуть быстрее.

CrossOver работает с «капсулами» [bottles], которые можно рассматривать как экземпляры рабочей среды Windows, настроенные под приложения, работающие внутри. Капсулы можно клонировать и сохранять (правда, кое-какие мелочи вроде шрифтов и прокси при этом теряются).

> «Codeweavers всегда сообщает, какие приложения ладят с его ПО.»

В десятой версии *CrossOver* эта идея получила дальнейшее развитие в виде системы CrossTie: теперь одним щелчком можно загружать готовые «рецепты» для приложений Windows, создающие лучшую среду для их установки. Таких программ около 500, и вы почти наверняка найдёте интересующее вас приложение. Мы пробовали эту функцию на различных программах, включая *Office 2007*, и всё работало безупречно. *CrossOver Pro* подготавливал рабочую среду, затем инсталлировал приложение с диска.



› **Если необходимо 100% совместимости с новейшими форматами *Microsoft Office*, выручит *Crossover Pro*.**

Немного Wine

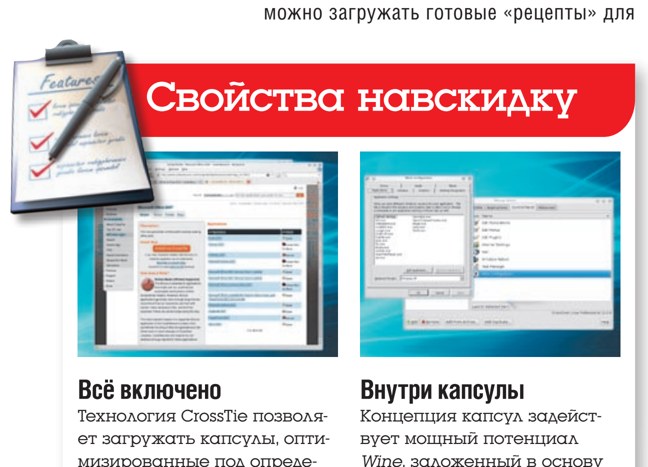
Наиболее значительной проблемой при работе с *CrossOver* может стать сама основа, *Wine*: его разработка не успевает за развитием ПО. Например, программисты сделали скачок в поддержке *Photoshop*, затем остановились на *CS2* – а ведь текущая версия уже *CS5*!

К счастью, Codeweavers всегда открыто сообщает о том, какие приложения ладят с его ПО, и гадать вам не придется. Кроме того, пробная версия щедро позволяет устанавливать самые редкостные программы. А значит, можно сделать осознанный выбор – бросить всё, использовать *Wine* или поискать полный пакет эмуляции.

Итак, если ваши открытые программы не справляются с работой, попробуйте *CrossOver*. Это особенно полезно, если вы не в ладах с командной строкой или хотите размножить приложение на нескольких машинах клонированием капсул. Хотя количество поддерживаемых программ ограничено, все они работают как полагается. Кроме того, обмен опытом по созданию оптимальных условий для работы различных программ – весьма удачное решение.

Теперь слегка попридираемся. Хотелось бы получить расширенный доступ к настройке параметров системы – например, работе с прокси и отображению шрифтов. Использование прокси – застарелый недостаток, и хотя *CrossOver 10* прекрасно подключается через прокси при инсталляции (используя системные настройки), передача информации приложениям происходит только после добавления аргументов во время запуска. В целом же *CrossOver* изрядно облегчает жизнь, за что ему и спасибо. LXF

Свойства навскидку



Всё включено
Технология CrossTie позволяет загружать капсулы, оптимизированные под определённые приложения.

Внутри капсулы
Концепция капсул задействует мощный потенциал *Wine*, заложенный в основу *CrossOver*.

Linux Format Вердикт

CrossOver Professional 10

Разработчик: Codeweavers
Сайт: www.codeweavers.com
Цена: £25,99 — стандарт; £44,99 — про

Функциональность	7/10
Быстродействие	8/10
Удобство использования	8/10
Оправданность цены	7/10

» Прекрасная альтернатива ограниченному Wine и сложным комплектам виртуализации.

Рейтинг 8/10

OpenSUSE 11.4

Антон Резник уверен, что новый openSUSE 11.4 просто обязан утешить расстроенных фанатов зеленого хамелеона (или геккона?!).

Вкратце

» **Один из основных дистрибутивов Linux. См. также:** *Fedora, Mandriva, Ubuntu*.

Ключевое ПО

» Ядро 2.6.37
» KDE 4.6
» GNOME 2.32
» LibreOffice 3.3.1
» Firefox 4.0
» GCC 4.5

Проигнорировав все предсказания анонимных аналитиков из Интернета, проект openSUSE предоставил свой обновленный дистрибутив. Продажа Novell, надо сказать, не на шутку взбудоражила общественность, так что все уже начали сильно сомневаться в дальнейшем продолжении разработки openSUSE. Но Attachmate, чье предложение было принято Novell (подробнее в **LXF139/140**), заявила, что проект будет жить. Заручившись таким обещанием, сообщество продолжило свою работу.

Точно в срок новый openSUSE 11.4 уже красовался на зеркалах проекта. Изменения новой версии заключаются не только в обновлении программного обеспечения и смене дизайна. Нет, конечно, последние стабильные версии популярных программ присутствуют, да и внешне openSUSE стал выглядеть более стильно. Но кроме этого, множество работы было проведено внутри системы.

К примеру, в стандартную поставку openSUSE уже включен нашумевший 200-строчный патч планировщика ядра. И он действительно работает! Еще предварительные релиз-кандидаты показывали заметные на глаз улучшения производительности. Другими словами – openSUSE работает так быстро, что у вас не будет даже времени подумать об отдыхе!



› **OpenSUSE решился на отважный шаг: *OpenOffice.org* заменен на его ответвление *LibreOffice*. Но все так же зеленеет.**

Движение вперед

Любители KDE определенно будут рады, ведь openSUSE 11.4 включает последний стабильный KDE 4.6 для настольного компьютера или маленького рабочего нетбука. Так как проект openSUSE делает основной упор на использование KDE в качестве стандартного окружения, то его исполнение здесь безупречное.

Последний стабильный GNOME 2.32 уже здесь! Несмотря на то, что почти все разработчики GNOME заняты подготовкой к выходу GNOME 3, вторая ветка этого рабочего окружения все еще продолжает обновляться и поддерживаться, предоставляя и последние версиивключенных в нее программ. К слову сказать, разрабатываемая 3-я ветка GNOME еще не может претендовать на статус стабильной, поэтому с openSUSE не поставляется. Однако любители новизны могут оценить возможности GNOME Shell, скачав специальный LiveCD с сайта SUSE Studio.

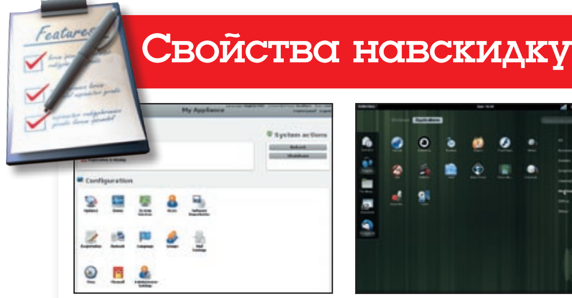
Уже собравшее некоторое количество поклонников крайне легковесное окружение *LXDE* окончательно вписалось в меню выбора DE при установке openSUSE. На фоне таких гигантов, как KDE и GNOME, *LXDE* смотрится очень уверенно и по-прежнему продолжает отвоевывать место на ваших компьютерах пользователей Linux. Кажется, настало время попробовать, на что он способен!

Кроме этого, в openSUSE 11.4 дебютирует *LibreOffice*. OpenSUSE – первый стабильный дистрибутив, поставляемый с новым свободным офисным пакетом. Это вполне объяснимо, ведь, разжившись со времени ответвления 133 новыми хакерами, *LibreOffice* уже хвастается своими личными наработками.

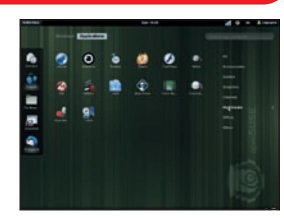
Установкой и настройкой системы по-прежнему руководит *YaST*. По сравнению с предыдущей версией, 11.3, установка проходит заметно быстрее.

Новые идеи насчет возможностей openSUSE получает с помощью системы предложений openFATE. Перед выходом 11.4 идей было так много, что наверняка разработчикам некогда было даже просто помечать. LXF

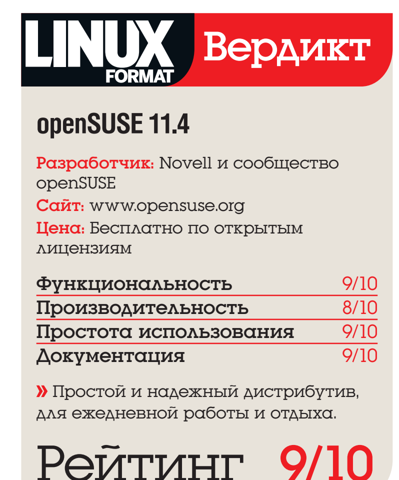
Свойства навскидку



WebYaST
Настройка всего и вся в openSUSE 11.4 доступна прямо из браузера, хотя пока портированы не все модули обычного *YaST*.



Gnome 3
Новый Gnome Shell уже можно попробовать. Правда, по умолчанию его в поставке нет.



LINUX FORMAT Вердикт

openSUSE 11.4

Разработчик: Novell и сообщество openSUSE
Сайт: www.opensuse.org
Цена: Бесплатно по открытым лицензиям

Функциональность	9/10
Производительность	8/10
Простота использования	9/10
Документация	9/10

» Простой и надежный дистрибутив, для ежедневной работы и отдыха.

Рейтинг 9/10

LibreOffice 3.3



Похоже, отпрыск *OpenOffice.org* затеял стать главным офисным комплектом для основных дистрибутивов. Хватит ли у него пороху, думает **Маянк Шарма**.

Вкратце

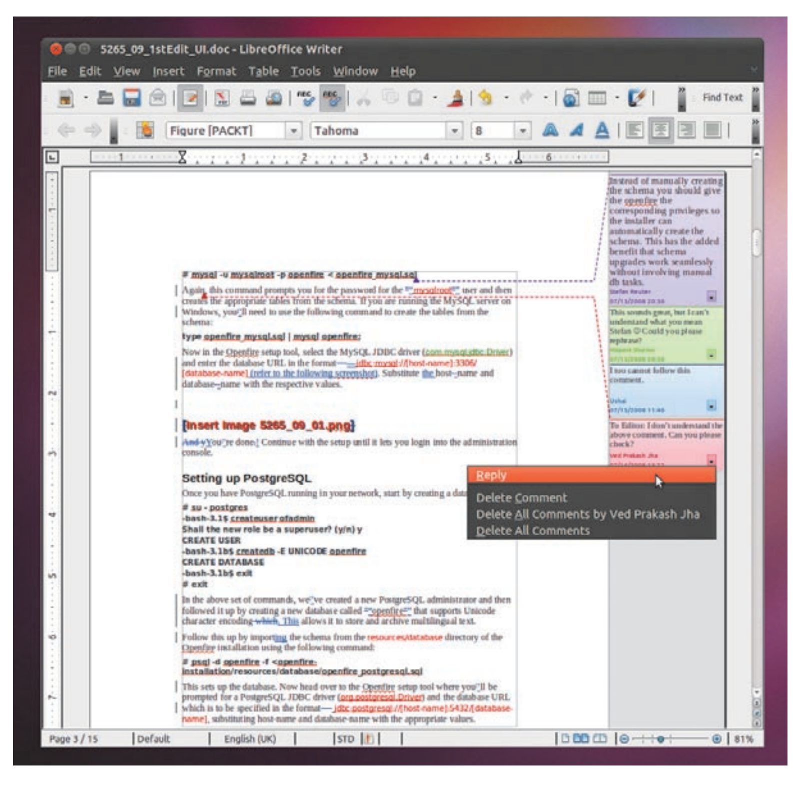
» **Офисный комплект, отпочковавшийся от одного из флагманов свободного ПО, с солидной поддержкой сообщества. См. также:** *KOffice*, *OpenOffice.org*.

Напрасно в Oracle не учли мнение Сообщества. Компания не сделала ничего, чтобы унять страхи по поводу судьбы *OpenOffice.org* в свете кончины OpenSolaris. С целью защитить разработки сообщества, несколько разработчиков покинули проект *OpenOffice.org* и сформировали организацию Document Foundation. Меньше чем за четыре месяца новая группа создала «свободную от Oracle» версию одного из наиболее узнаваемых брендов открытого ПО.

LibreOffice принял эстафету у *OpenOffice.org*, причём во многих смыслах. Хотя это была первая версия группы, нумерацию начали с *LibreOffice 3.3* (на время разделения рабочая версия *OpenOffice.org* имела такой же номер).

Разрешение конфликтов

Загрузить *LibreOffice* можно с сайта www.libreoffice.org, а можно взять его из репозиториев основных дистрибутивов, например Ubuntu и Fedora. Но будьте внимательны: когда мы устанавливали *LibreOffice* с PPA Ubuntu, новичок первым делом укокошил *OpenOffice.org*. Хотя в официальном руководстве по установке *LibreOffice* о таком удалении ничего не говорится, на практике два комплекта не очень-то дружат – во время бета-тестирования многие пользователи сообщали о проблемах с конфликтами зависимостей. Согласно официальным сведениям о данной версии, *LibreOffice* неустойчиво работает с вариантом Java *GCJ*, поэтому рекомендуется использовать *OpenJDK*. Загружая комплект непосредственно с сайта, придётся отдельно брать пакеты для различных языков и справочные файлы.



› **Выглядит впечатляюще: *LibreOffice 3.3* делает серьезную заявку.**

На первый взгляд *LibreOffice* выглядит и действует почти идентично *OpenOffice.org*. Кроме нескольких косметических штрихов, основная работа над этой версией проходила подспудно (отладка кода, исправление ошибок и т.п.). Поскольку разработчики *LibreOffice* приступали к тому, что должно было стать *OpenOffice.org 3.3*, в новой версии можно найти все функции, характерные для новейших версий *OOo*. В том числе это новый шрифт Liberation Narrow, возможность встраивать стандартные шрифты PDF в соответствующие документы, усиленная защита документов *Writer* и *Calc* и способность создавать таблицы *Calc* с количеством строк до миллиона.

Внешний вид

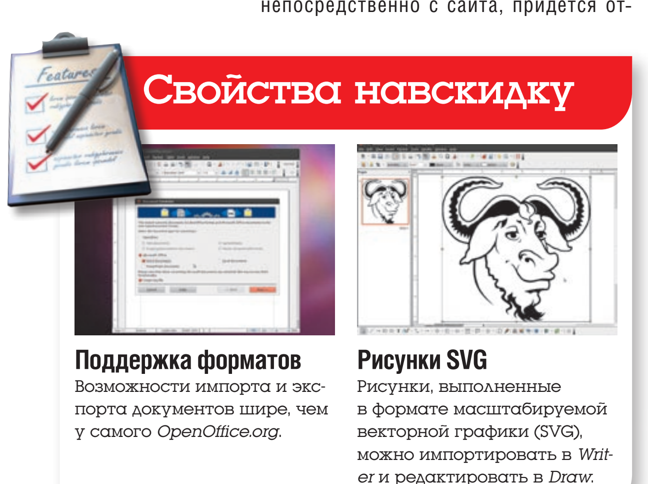
Косметические перемены коснулись интерфейса пользователя, например, управления компоновкой слайдов в *Impress*; в диалоговом окне «Сохранить как...» отображаются все доступные форматы файлов для сохранения; переработан интерфейс печати и встроенный инструмент «Тезаурус».

Встречали мы версии *OOo* с меньшим количеством видимых и подспудных улучшений. Вероятно, сообщество взликовало бы и при простом дублировании функциональности – но, не почив на чужих лаврах, разработчики *LibreOffice* ввели в свой комплект функции, превосходящие достижения предка-конкурента.

Например, в *LibreOffice 3.3* есть экспериментальный режим (Сервис > Параметры > Общие) для тестирования функций, находящихся в разработке. Сейчас с помощью этой функции возможно непосредственное редактирование формул (то есть ввод символов непосредственно в формулы *Calc*), а также запись макросов – многолетняя ахиллесова пята *OOo*.

Кроме того, *LibreOffice* также предоставляет ряд дополнений к интерфейсу. Тот, кому приходилось создавать многостраничные отчёты в *Writer*, оценит новое диалоговое окно для создания и редактирования титульных страниц (откры-

Свойства навскидку



Поддержка форматов

Возможности импорта и экспорта документов шире, чем у самого *OpenOffice.org*.

Рисунки SVG

Рисунки, выполненные в формате масштабируемой векторной графики (SVG), можно импортировать в *Writer* и редактировать в *Draw*.

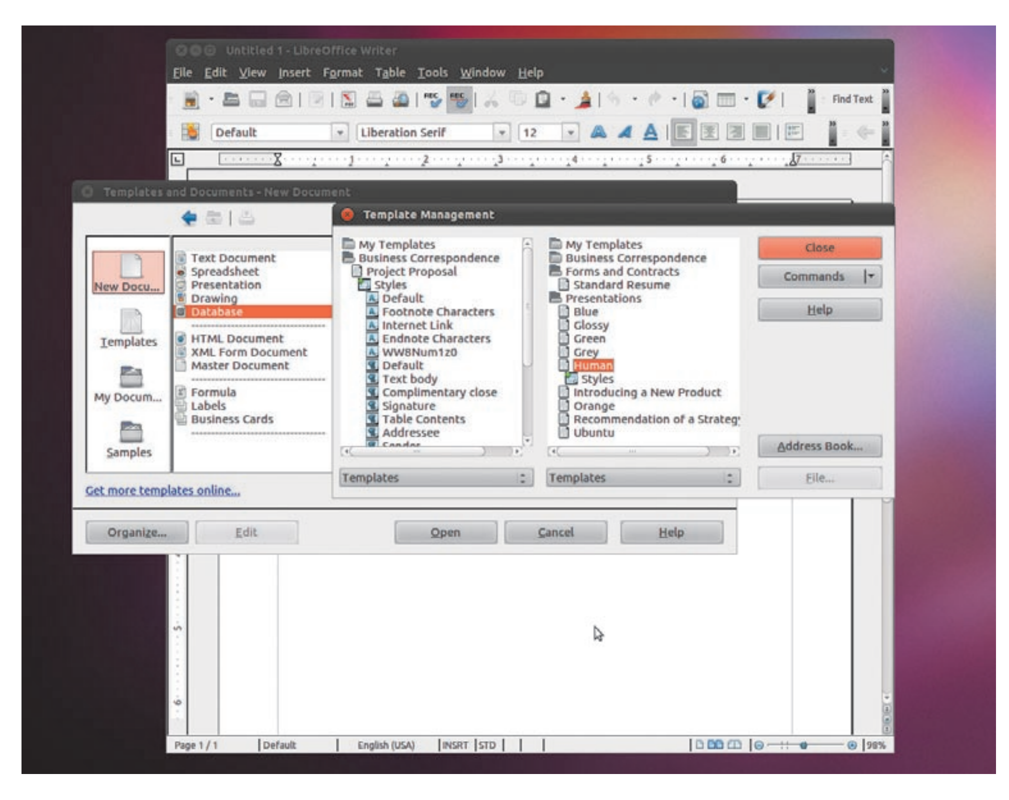
вается при выборе пункта «Титульная страница» в контекстном меню). В *Calc* появилась новая страница параметров совместимости, с которой можно быстро переключиться на прежние клавиатурные комбинации *OpenOffice.org*. Если пакет справочных файлов не установлен, функция LibreOffice Help откроет для вас онлайн-систему справочных страниц wiki.

Рука помощи

Мощная поддержка *LibreOffice* была оказана *Go-oo* (ответвление *OpenOffice.org* под опекой Novell). Фактически Novell свернула разработку *Go-oo* в пользу работы с Document Foundation. Благодаря вкладу Novell, *LibreOffice* может импортировать документы *Lotus Word Pro* и *Microsoft Works* с помощью библиотеки *libwps*. По словам разработчиков, улучшен также импорт документов *Wordperfect*.

Кроме того, в комплект можно импортировать (а затем обрабатывать) файлы изображений, созданные в формате масштабируемой векторной графики (SVG). Их можно импортировать в *Writer* и редактировать в *Draw*. Понятно, что эта функция понадобится лишь малой части пользовательской базы; тем не менее она существует, и при необходимости к ней можно прибегнуть.

Одно из наиболее значительных последствий сотрудничества с *Go-oo* – надежная поддержка OOXML. В *LibreOffice* введены заплаты для улучшенной поддержки документов *Microsoft Office 2007* и *2010*. Согласно заявлению Document Foundation, они были «...результатом особого соглашения между Novell и Microsoft».



› В *LibreOffice* полно мастеров, шаблонов и расширений — а если вам мало, загрузите с сайта еще.

Вдобавок *LibreOffice* позволяет даже сохранять документы OOXML. Хотя инициативы Novell по поддержке OOXML одобряют не все члены сообщества, *LibreOffice* вносит свою лепту в пропаганду открытого формата документов (ODF), предлагая использовать его вместо сохранения в OOXML.

Судя по первой версии, *LibreOffice 3.3* – полностью самостоятельное явление. Однако было бы неверно считать его совершенно новым проектом. В сущности, это ответвление одного из столпов открытого ПО, чьи корни уходят в глубь прошлого столетия. Если рассматривать событие под таким углом, то это очень напоминает очередной релиз. Правда, пока *LibreOffice 3.3* располагает не всеми инструментами, которые приличествуют современному офисному комплекту. Например, здесь нет средств для совместной работы в «облачном» формате.

Радует то, что Document Foundation удалось собрать кадровый потенциал для интенсивной новаторской работы, а также заручиться мощной общественной поддержкой и помощью ведущих отраслевых компаний. *LibreOffice 3.3* может стать отличной площадкой для разработки полноценной альтернативы проприетарным небожителям типа *Microsoft Office* и *Google Docs*. LXF

Преимущество LibreOffice

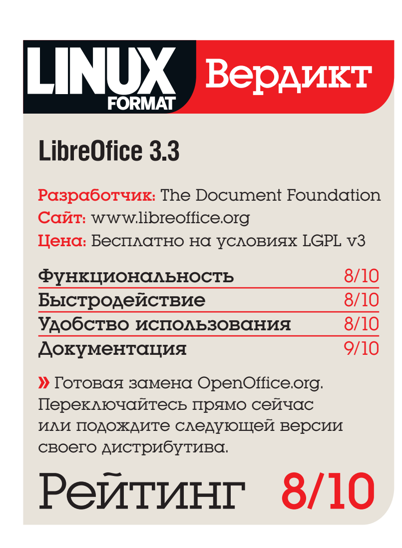
Призывы передать разработку *OpenOffice.org* под серьёзную опеку раздавались задолго до того, как Oracle приобрела Sun. Поглощение и прекращение OpenSolaris послужили катализатором процесса.

Но даже под управлением Sun модель разработки и выпуска версий следовала прежде всего потребностям частной организации. «Новые версии выходили нечасто, а наложение заплат происходило очень медленно из-за особенностей циклов контроля качества Sun, проблем формирования системы и других бюрократических препон, вроде соблюдения лицензий на авторские права», рассказывает Рахул Сундарам [Rahul Sundaram], ответственный за формирование пакетов Fedora.

Организация Document Foundation была основана в конце сентября 2010 г. горсткой бывших разработчиков *OpenOffice.org*, отделившихся от Oracle; теперь же в проект влилось свыше 100 человек. *LibreOffice 3.3* – это результат четырёхмесячного труда по отладке кода, внедрению функций и бета-тестированию.

Сундарам добавляет, что в Fedora и других дистрибутивах уже есть некоторые из новых функций *LibreOffice*, поскольку на пакеты *OOo* из «стволовой» версии Oracle накладывались заплаты, полученные от *Go-oo* при поддержке Novell.

Интересный момент: позиция Oracle в отношении вклада третьих сторон может сыграть на руку разработчикам *LibreOffice*. По условиям GPL, любое изменение, которое Oracle вносит в *OpenOffice.org*, тут же становится достоянием гласности, и разработчики *LibreOffice* могут при необходимости включить его в свой комплект. А вот для *OOo* такой номер не проходит, поскольку Oracle соблюдает все условия лицензионных соглашений – чтобы не лишиться права продавать собственную проприетарную версию. «Вот так у *LibreOffice* образовалось стратегическое преимущество», говорит Сундарам.



LINUX FORMAT **Вердикт**

LibreOfice 3.3

Разработчик: The Document Foundation
Сайт: www.libreoffice.org
Цена: Бесплатно на условиях LGPL v3

Функциональность	8/10
Быстродействие	8/10
Удобство использования	8/10
Документация	9/10

» Готовая замена OpenOffice.org. Переключайтесь прямо сейчас или подождите следующей версии своего дистрибутива.

Рейтинг 8/10

Сравнение



Каждый месяц мы сравниваем тысячи программ — а вы можете отдыхать!

LINUX FORMAT на DVD
» 4Pane 0.8.0
» PcManFM 0.9.9
» Rox-Filer 2.10

Файловые менеджеры

Растет размер жесткого диска – растет и количество файлов. Для тех, у кого оно уже зашкалило, **Марко Фиоретти** нашел шесть программ на подмогу.



Про наш тест...

Все файловые менеджеры, описанные на следующих четырех страницах, тестировались на двуядерной системе AMD 64 $5800+ с 8 ГБ ОЗУ. Система работала на Fedora 14 со всеми обновлениями, доступными к январю 2011. За исключением *Rox-Filer*, все программы были установлены из RPM-пакетов, и они также имеются в форматах TGZ и Deb для разных дистрибутивов. *Rox-Filer* выпал из этого ряда, поскольку у него нет свежих RPM. Вместо них он использует другой метод установки (см. обзор, где даны все подробности). Проведенные тесты в основном состояли в перемещении, переименовании и поиске по большому количеству файлов и папок самых разнообразных видов. Среди них была папка, содержащая почти 240 цифровых изображений, которые вы увидите на некоторых экранных снимках.

Наша подборка

4Pane с. 19
Dolphin с. 18
Nautilus с. 17
PcManFM с. 20
Rox-Filer с. 20
Thunar с. 19

Среднее число файлов, хранящееся на настольном ПК, постоянно растет. Наши потребности меняются вместе с нами – с каждым годом мы все больше зависим от цифровых документов, и наши устройства хранения увеличиваются, чтобы не отставать. Короче говоря, способ организации файлов, которым мы гордились еще пару лет назад, уже устарел. Вывод – вам нужен не просто эффективный помощник для повседневной работы с файлами, но также и тягловая лошадка для серьезного перепахивания диска.

Поэтому даже в наше время социальных рабочих столов и облачных вычислений файловые менеджеры все еще актуальны. И лакмусовый тест для любого файлового менеджера – способность эффективно управлять множеством файлов; это один из двух основных критериев для приложений в Сравнении. Другой важнейший для нас критерий – убедительность. Сможет ли каждый из представленных здесь файловых менеджеров содействовать в задаче убеждения неопытного пользователя Linux в том, что эта ОС может быть и привычна, и проста в использовании, но отличаться от того, чем он пользовался раньше, большей гибкостью?

В итоге мы отвели много места файловым менеджерам популярных и дружелюбных к новичкам сред рабочего стола. Заодно включены те, что имеются в большинстве live CD Linux для систем с низкими требованиями к аппаратуре, и пара аутсайдеров, чтобы представить весь спектр экосистемы Linux. Понятно, мы могли бы включить для этого и ценные программы – например, *Midnight Commander* или *EmelFM2*, но они недавно рассматривались в **LXF**, и мы решили на сей раз обойтись без них.

> «Способ организации файлов, которым мы гордились еще пару лет назад, уже устарел.»

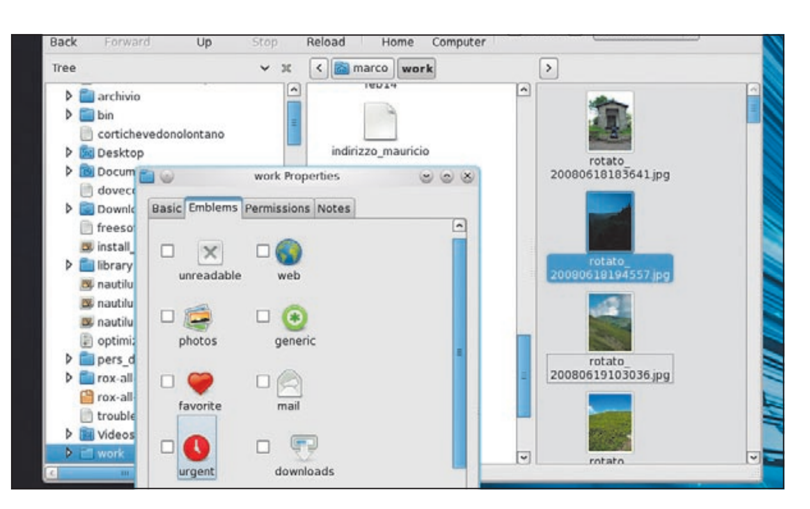
Nautilus

Файловый менеджер Gnome – по-прежнему эталон для многих других проектов.

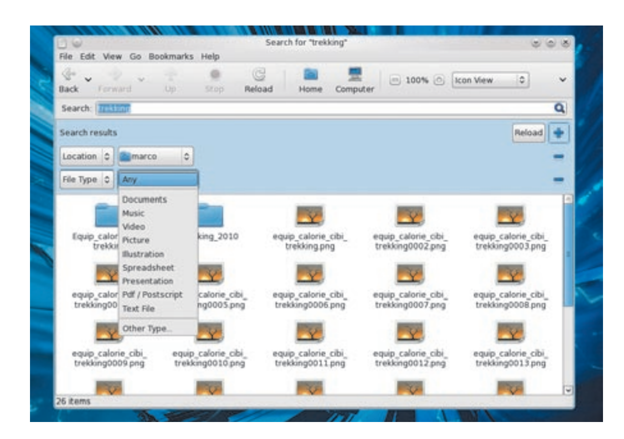
Запустите *Nautilus* в Fedora 14, и по умолчанию вы увидите многопанельную структуру с деревом слева, несколькими значками вверху и солидным пространством для просмотра файлов. Это режим браузера [Browser mode]. Альтернатива – «просторный» режим [Spatial mode]: окно открывается для каждой папки, куда вы получаете доступ – почти как если бы каждая была физической папкой. Установите, что вам нравится, по умолчанию через меню Preferences приложения.

Но возможности этим не исчерпаны: *Nautilus* умеет отображать содержимое папки в виде List [Список], Icons [Значки] или Compact [Компактное отображение]; все они доступны из выпадающего меню. Можно назначить для открытия файла одиночный или двойной щелчок, а уменьшенные изображения отключать, включать и увеличивать. Масштаб меняют две маленькие кнопочки по бокам уровня масштаба. Каждый щелчок по ним отвечает 50 % изменению, но мы бы предпочли более тонкую настройку. Более того, можно дополнительно просматривать несколько папок в одном окне *Nautilus*. Новая панель открывается по выбору View > Extra Pane.

В нашем виде по умолчанию Browser *Nautilus* отображает текущую директорию с навигационным путем – то есть с указанием пути в текущую директорию, который вы можете использовать, чтобы отследить свои действия. Правый щелчок по нему дает выбор: идти этим путем дальше или открыть папку в отдельном окне или вкладке. Другие кнопки на верхней панели позволяют перемещаться по недавно открытым папкам или перейти на уровень вверх в файловой системе. Причем неважно, локальные это файлы или удаленные: *Nautilus* умеет соединяться с серверами FTP, WebDAV, SSH и ресурсами разделенного доступа Windows. Можно также сохранять ранее открытые папки в виде закладок.



› *Nautilus* снабжен простой системой Emblem, дающей визуальные подсказки о категории, к которой принадлежит файл или папка.



› *Nautilus* может осуществлять поиск внутри папок по ряду критериев, в том числе по тэгам.

> «Emblems – визуальные напоминания о целевом назначении файла.»

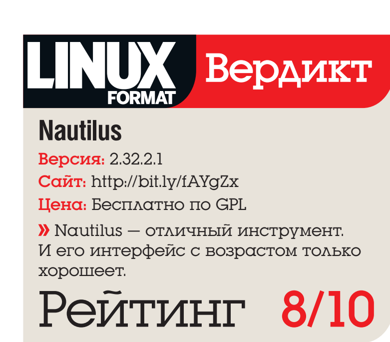
Маленькая стрелка в правом конце панели инструментов активирует функцию поиска *Nautilus*. В ней можно определить, какие виды файлов искать и где. Вы даже можете задать поиску имя и добавить его в закладки, как для обычной папки.

Чтобы помочь вам с ходу отыскивать файлы, *Nautilus* может прикреплять к вашим файлам Notes [Примечания] и Emblems [Эмблемы] – небольшие картинки, добавляемые к значку файла. Для добавления Note, щелкните правой кнопкой по файлу, выберите Properties, перейдите во вкладку Notes, напечатайте, что хотели, и нажмите Save. После этого ваш текст станет видимым в левой панели при каждом включении режима Notes Display. А Emblems можно применять как визуальные напоминания о назначении или природе каждого файла: например, добавить эмблему Важности особо ценным файлам. Если предопределенные здесь опции не соответствуют вашим нуждам, создайте собственные: перейдите в Edit > Background And Emblems, затем щелкните по Emblems и Add A New Emblem. К сожалению, нет явной возможности использовать Emblems как критерии поиска.

И, наконец, при выборе файла или папки *Nautilus* открывает контекстно-чувствительное меню, где всегда есть опция Send To, пригодная для отправки файлов по системе обмена быстрыми сообщениями или по электронной почте, на DVD или в программы сжатия. Простой способ расширить способности *Nautilus* – добавить в меню собственные действия. Действия *Nautilus* – это просто исполняемые скрипты оболочки, помещенные в специальную директорию, обычно **$HOME/.gnome2/nautilusscripts/**. Ее содержимое отображается в подменю Scripts. При выборе скрипта *Nautilus* запускает его с выбранными файлами в качестве параметров. Этих скриптов уже немало накопилось онлайн: вы найдете их на http://g-scripts.sourceforge.net или по заголовку http://bit.ly/fZs4uP.

Сомнительное дело

Несколько лет назад *Nautilus* сочли достаточно громоздким и неуклюжим, чтобы он подлежал замене. Так возник *Nautilus Elementary*. В конце 2010 этот проект был закрыт, и его заменил совершенно новый файловый менеджер под названием *Marlin*. Пока рано говорить о том, превзошел ли он *Nautilus*, особенно сейчас, когда интерфейс и настройки по умолчанию псевдо-головоногого менеджера стали проще и эффективнее; однако на *Marlin* стоит обратить внимание в этом году.



LINUX FORMAT Вердикт

Nautilus

Версия: 2.32.2.1
Сайт: http://bit.ly/fAYgZx
Цена: Бесплатно по GPL

» Nautilus – отличный инструмент. И его интерфейс с возрастом только хорошеет.

Рейтинг 8/10

Dolphin

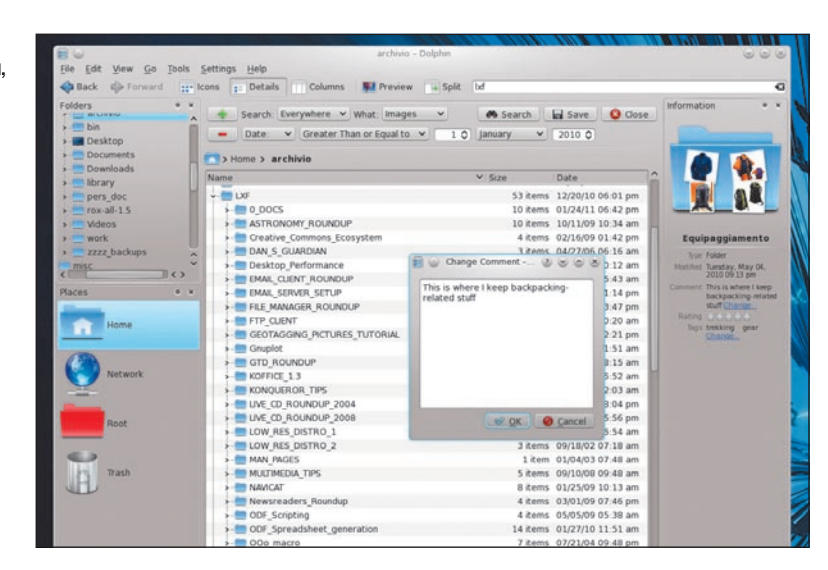
Мощный, гибкий инструмент с двумя полунезависимыми режимами поиска.

Dolphin – официальный файловый менеджер рабочего стола KDE. Его стандартная конфигурация дает пользователю левую панель с листингом Places, с Home, Network, Root и Trash в качестве опций по умолчанию. Основная область служит для отображения содержимого текущей директории в виде Icons [Значков], Columns [Столбцов] или Details [Подробно]. Последнее напоминает листинг в окне терминала. При его использовании можно показать или скрыть иерархию дерева директории с помощью опции Expandable Folders. Кнопка Split [Разделение] включает или отключает двухпанельный вид; а можно открыть несколько вкладок. Если вам неизвестна функция кнопки, наведите на нее курсор, нажмите на Shift+F1, и получите подсказку.

Имеются пиктограммы, что является вполне стандартным. Функцию изменения масштаба в *Dolphin* заметить труднее, чем в *Nautilus*: это всего лишь крошечный слайдер в нижнем правом углу. Однако он работает лучше, поскольку настраивается на любой шаг, соответствующий предварительно заданной шкале.

С помощью View > Location можно настроить панель местонахождения в режиме Breadcrumb или Editable Box mode. Почти все горячие комбинации клавиш в *Dolphin* настраиваются, как и приоритет критерия сортировки файлов. *Dolphin* также имеет панели для Information [Информации] и Folders [Папок] и интегрированный терминал. Панели можно максимизировать, минимизировать и отделять от главного окна *Dolphin*, нажав на маленькую ромбовидную кнопку в углу панели.

› Интерфейс не загромождается, даже если открыть все его части, но вряд ли они вам понадобятся все сразу.



Панель Information предоставляет предпросмотр выбранного в основной области. Возможно, это также самая сильная сторона *Dolphin*, потому что позволяет прикреплять комментарии, рейтинги и тэги к каждому файлу или папке в менеджере (или, скорее, в *Nepomuk*) и отображает все введенное ранее. Механизм подобен менеджеру изображений *Digikam*; *Dolphin* просто обобщил этот подход и интерфейс. Возможно, он требует времени, но результат стоит того, если у вас много файлов.



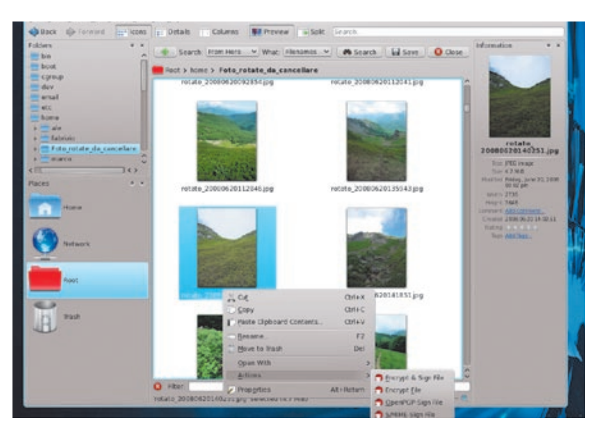
› *Dolphin* — это не *Digikam*, но подходит для быстрой черновой реорганизации изображений.

> **«Split включает или отключает двухпанельный вид, или вкладки.»**

По части нахождения файлов, поисковые способности *Dolphin* мощны, но способ их презентации может обескуражить начинающих пользователей. Есть две разных функции поиска, но их отличие не так очевидно, как следовало бы. Один интерфейс представляет собой окно, которое выводится при нажатии на Ctrl+F или выборе Tools > Find File. Другой – небольшое поле ввода, появляющееся в правом конце панели инструментов при выборе Settings > Toolbars Shown. Окно Find File позволяет осуществлять поиск в традиционном для Unix смысле: в соответствии с именами файлов, содержанием, если это текст, или такими свойствами, как размер или дата изменения.

Другой интерфейс делает нечто совершенно иное. Вы так же можете уточнять результаты, задав ограничения по размеру файла или дате изменения. Однако эта панель инструментов отправляет запрос в базу данных *Nepomuk*. Если активировать его индексатор, он будет просматривать также и содержание файлов, но в противном случае вернет файлы, чьи тэги или комментарии соответствуют тому, что вы ввели в окно Information. Как и в *Nautilus*, вы можете сохранять запросы в форме закладок, и в *Dolphin* тоже есть настраиваемое контекстное меню.

Помимо поддержки информации о файле, еще одной сильной стороной *Dolphin* является наличие интересных примочек самых разнообразных типов, в том числе интерфейса Dropbox, создания DVD или галерей изображений с помощью *Kdenlive* и конвертирования сканированного текста в простой текстовый формат с помощью *Tesseract*.



Семантический десктоп

Интеграция *Dolphin* с *Nepomuk* – практическая демонстрация того, как выглядит семантический рабочий стол. Однако помните, что индексатор *Nepomuk*, *Strigi*, может затормозить работу старого компьютера. К счастью, его легко отключить в панели System Settings > Desktop Search в KDE, если это проблема.

LINUX FORMAT Вердикт

Dolphin

Версия: 1.5
Сайт: http://dolphin.kde.org
Цена: Бесплатно по GPL

» Интерфейс Dolphin опрятен, и при этом вмещает массу функций.

Рейтинг 9/10

4Pane

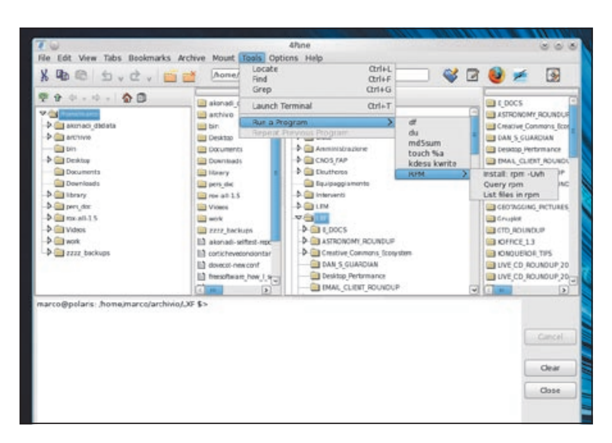
Спартанская, но эффективная рабочая лошадка; с ней легко.

Этот многопанельный списочный файловый менеджер был разработан, чтобы отдать предпочтение скорости перед визуальными эффектами. Его название вызвано тем фактом, что по умолчанию отображаются две идентичных пары панелей; каждая из пар состоит из древовидной директории и списка файлов. Но можно добавлять и удалять панели, применять вкладки и сохранять закладки.

При первом запуске *4Pane* мастер изучит вашу систему, чтобы создать базовую конфигурацию или позволить импорт готовой. Подогнав *4Pane* к своему вкусу, сохраните эту настройку как шаблон. Для большинства функций есть комбинации клавиш, и их можно настраивать – имеющаяся в *4Pane* «Подсказка для тех, кто не читает руководств» гласит: «Настраивается почти все, в основном через меню Options > Configure 4Pane; нажав на F1, вы увидите контекстно-зависимую подсказку».

Интерфейс содержит кнопки или меню для монтирования и отмонтирования приводов или разделов диска, управления архивами и возврата к недавно посещенным папкам. Кроме того, *4Pane* поддерживает массовое переименование или дублирование файлов, и работает с внешними дисками, ресурсами NFS или *Samba*. Следуя подробным инструкциям в полном руководстве пользователя, в *4Pane* можно добавить определенные пользователем инструменты. Что там есть, покажет меню Tools – отобразится то, что поддерживается в 4Pane: **find**, **grep** и **locate**.

В *4Pane* даже интегрирован терминал. Он открывается с помощью View > Show Terminal Emulator. Правда, нельзя использовать **su**, **sudo** или управляющие последовательности. Зато поддерживаются «канализация» [pipes], переадресация [redirection] и история команд [command histories], что экономит время, особенно если вы хотите работать наполовину мышью, наполовину из командной строки.



› ***4Pane* позволяет легко сравнивать разные части вашей файловой системы.**

> «Подогнав 4Pane к своему вкусу, сохраните настройку как шаблон.»



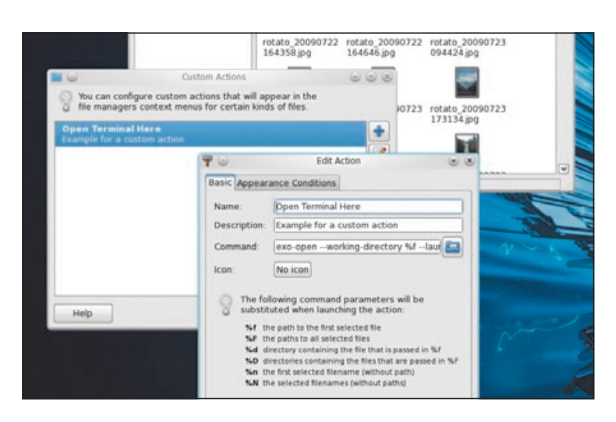
Thunar

Файловый менеджер выбора для многих нетбуков и легких приложений.

Thunar – это файловый менеджер по умолчанию рабочего стола *Xfce4*. Он прост, но хорошо продуман, быстр и удобен. В главном меню – только пункты File [Файл], Edit [Редактировать], View [Просмотр] и Go [Перейти]. Закладки и подмонтированные приводы отображаются в левой панели в виде значков или в древовидной структуре, а текущая папка – как навигационная цепочка в стиле Pathbar *Thunar* либо в редактируемом текстовом окне в стиле Toolbar. Последний также добавляет кнопки Up, Back и Home, аналогичные кнопкам *Nautilus*. Еще одна аналогия – выбор между детальными значками и компактным либо подробным списком для отображения.

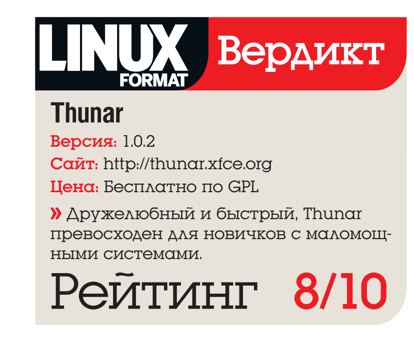
Команда Open Terminal Here [Открыть терминал здесь] всегда доступна в главном контекстном меню. Опция Send To [Отправить], там же, может создать ссылку на рабочем столе для выбранного файла, прикрепить его к почтовому сообщению или скопировать на другие приводы или устройства Bluetooth. Последняя опция – массовое переименование, активируемое также нажатием F2 или выбором Edit > Rename… в главном меню. Здесь можно добавлять, удалять или перемещать символы в именах выбранных файлов или приписывать даты либо возрастающие номера. Дополнительные опции переименования доступны для определенных типов файлов через модули расширения – список модулей Действий есть на http://bit.ly/f1wUYp.

Создать свое Действие тоже несложно. Выбрав Edit > Configure Custom Actions и щелкнув по значку с плюсом, вы получите панель с двумя вкладками. Первая сообщает *Thunar*, как присоединить некую программу к новому пункту меню. Вторая определяет Appearance Conditions: типы файлов, с которыми *Thunar* должен ассоциировать этот пункт меню.



› **Если вам мало основных функций *Thunar*, добавьте свои.**

> «Thunar прост, но хорошо продуман, быстр и удобен.»



Rox-Filer

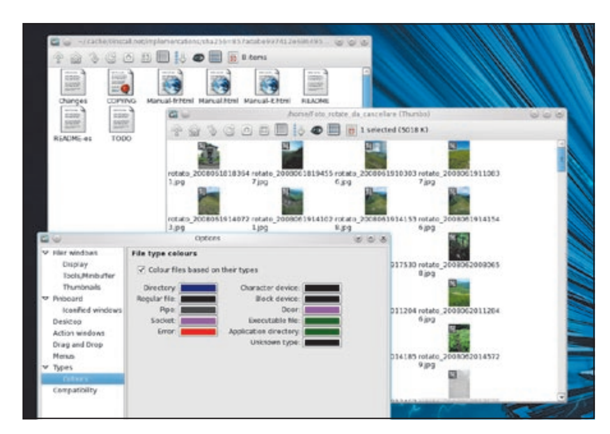
RISC живет в Linux. К сожалению, пушки не задействованы.

Rox-Filer – всего лишь часть Rox, рабочего стола, вдохновленного стратегией RISC (Reduced Instruction Set Computing – вычисления с ограниченным набором команд). В окне *Filer* всего несколько кнопок. Три – это ярлыки для закладок, каталога **home** и родительской директории. Прочие сортируют, скрывают или показывают файлы или отображают миниатюры, а не значки. Файлы можно раскрасить в зависимости от их типа.

Функции у *Rox-Filer* почти те же, что и у других файловых менеджеров, но размещены в другом месте – а именно, во всплывающем меню, активируемом щелчком правой кнопки мыши. Отсюда вы можете открыть терминал или панель размещения, запустить поиск файлов (даже с регулярными выражениями), активировать отображение в виде List [Список] или Icon [Значки] и т. д. Подменю Next Click даже дает возможность выбрать следующее действие, прежде чем сообщить *Filer*, какой файл или папка станут объектом этого действия. Терминал по умолчанию здесь *Xterm*, но это можно изменить.

Рабочий стол Rox использует систему *0install*, которая размещает программы в независимых директориях приложений. Установка таких программ не сложнее копирования папки, что мы и делаем с помощью *Filer*. Полное руководство имеется и в формате HTML, и онлайн. Однако по щелчку на значке LIFESAVER (по крайней мере, в Fedora 14), *Filer* откроет не страницы своего руководства в браузере, а только папку, куда они были скопированы.

Работа с *Filer* интригует. Он быстр и гибок, и внешне не похож на других. Иногда начинает казаться, что вы вернулись лет на 10 назад, но в хорошем смысле – можно представить себе, каким бы стал Linux, если бы пошел другим путем. И хотя здесь есть все необходимые функции, помните, что для полного овладения ими вам придется потрудиться.



› **Интерфейс *Rox-Filer* выглядит голым, но он быстр, и в нем есть все, что вам действительно нужно.**

> «Кажется, что вы вернулись лет на 10 назад, но в хорошем смысле.»

LINUX FORMAT Вердикт

Rox-Filer

Версия: 1.5
Сайт: http://bit.ly/ijk3KT
Цена: Бесплатно по GPL

» Rox-Filer с виду незамысловат, но мощен и быстр: попробуйте!

Рейтинг 7/10

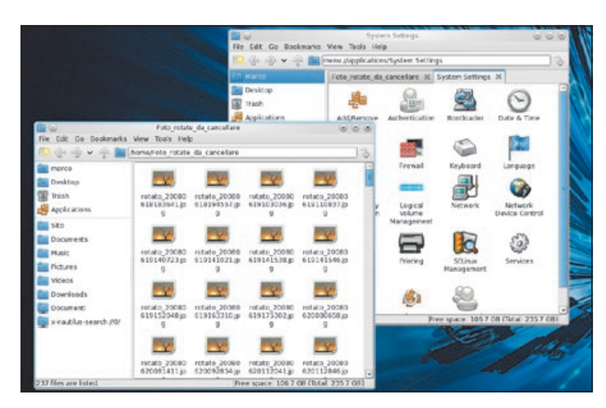
PcManFM

Файловый менеджер, столь простой, что не надо и документации?

PcManFM, вероятно, можно считать самым простым и наименее пугающим файловым менеджером на этих страницах. В панели инструментов под главным меню всего пять кнопок: для предыдущей и следующей папок, ссылок на родительскую и домашнюю директории и для открытия новой вкладки. Кроме них, имеются панель размещения и стрелочка, открывающая список недавно посещенных папок. Левая панель разделена на две части: графическую версию меню Go и Bookmarks. Часть этой панели Go содержит ярлыки для некоторых предварительно определенных размещений. По умолчанию это ваша домашняя директория, папки **Desktop**, **Trash** и **Applications**, которая позволяет запускать все приложения, управляющие настройками Gnome System.

В общем и целом, внешний вид и ощущения от работы с *PcManFM* напоминают нам сильно урезанную версию *Nautilus* в стиле *Thunar* – примите за комплимент. И это – только часть причин, по которым *PcManFM* применяется в LXDE, Light Desktop Environment. Эта программа быстра и легковесна, но притом предоставляет пользователю возможность просмотра во вкладках и 4 режима отображения: Icon [Значки], Thumbnail [Миниатюры], List [Список] или Compact [Компактное отображение]. Она также способна на массовое управление, и допускает перетаскивание между вкладками. Есть функция миниатюризации для файлов изображений, плюс закладки и работа с именами файлов в кодировках помимо UTF-8.

Основные ограничения данного приложения – отсутствии интерфейса поиска и только две возможности: открыть терминал в текущей папке (F4) или от имени пользователя root. Прямых ссылок на документацию нет вовсе, но интерфейс настолько прост, что это не проблема.



› **Трудно найти интерфейс проще и приятнее, чем у *PcManFM*.**

> «В панели инструментов под главным меню всего пять кнопок.»



Файловые менеджеры

Вердикт

Dolphin 9/10

Все представленные здесь файловые менеджеры хорошо работают на настольных машинах от среднего до высокого класса, ставших вполне обычными. Если вы ищете что-то простое в использовании на машине с ограниченными возможностями, хорошим выбором для вас станут как *Thunar*, так и *PcManFM*. Если простота и знакомый внешний вид для вас не основные критерии, попробуйте нечто новенькое – *Rox-Filer*, или *4Pane*, если вам приходится перемещать или анализировать множество неграфических файлов с помощью **find** и **grep**.

Однако в общем, полноценный файловый менеджер, пригодный для большинства пользователей, должен предлагать нечто большее, чем эти приложения. Сложное управление файлами требует некоторых ключевых функций – таких, как соединение графического интерфейса и быстрой командной строки в одно приложение как можно ровнее и как можно более гладко. Первый полезен, чтобы выяснить, является ли эта уйма файлов JPEG с непонятными названиями вашими отпускными фотками или графиками прогноза продаж. Второе удобно для тех моментов, когда требуется большая гибкость, автоматизированность или скорость, чем способна предложить программа только с GUI. Конечно, можно запустить файловый менеджер в одном окне и терминал в другом, но если они оба под рукой, работается быстрее. Еще одна важная функция – прямая поддержка поиска (нам очень нравится **find**, но он не универсален) и возможность расширения.

Вот поэтому мы рады объявить победителем этого Сравнения *Dolphin*, а второе место с небольшим отрывом занял *Nautilus*. *Dolphin* – отличный коктейль из всех перечисленных нами функций, и не только. Терминал в окне и интеграция с *Nepomuk* для семантического поиска – его главные козыри, и хотя система Emblems в *Nautilus* определенно удобна, мы считаем ее менее гибкой, чем информация о файле, с которой умеет работать *Nepomuk*. Однако оба – отличные приложения. Глядя на них, стоящих бок о бок, мы сожалели только об одном: почему они не делят метаданных файлов и, по крайней мере, их самых простых расширений? Будь мир совершенен, было бы идеально, если бы, например, *Nautilus* мог читать тэги *Nepomuk*, введенные из *Dolphin*, или если бы файловый менеджер KDE автоматически находил и применял скрипты оболочки *Nautilus*. LXF

› Мало того, что *Dolphin* многофункционален: поиск и установка модулей расширения в нем крайне просты.

Обратная связь

Вы знаете приложение, которое организовано даже лучше, чем Конвенция о полиции? Вам все еще не хватает каких-то убойных функций для управления файлами? Тогда шлите нам свое мнение по адресу letters@linuxformat.ru.

Ознакомьтесь с цифрами

Название	Инструментарий	Размер пакетов	Среднее потребление памяти	Настраиваемость	Альтернатива
4Pane	wxWidgets	3,8 МБ	20 МБ	Средняя	—
Dolphin	Qt	90 МБ [1]	30 МБ	Высокая	Nautilus
Nautilus	GTK	18 МБ	25 МБ	Высокая	Dolphin
PcManFM	GTK	0,7 МБ	12 МБ	Низкая	Thunar
Rox-Filer	GTK	1,2 МБ	11 МБ	Средняя	—
Thunar	GTK	13 МБ	13 МБ	Средняя	PcManFM

[1] Размер всего Kdebase, включая *Dolphin*.

Примечание: Приведенные размеры пакетов и статистика потребления памяти приблизительны. И то, и другое зависит от систем, конкретной настройки программ, выполняемой задачи и многих других факторов. Это особенно касается *Dolphin*, который еще и взаимодействует с другими компонентами типа *Nepomuk* и приложениями, доступными в двоичном формате только в связке с другим ПО.



Выстругайте себе посохи, стряхните пыль с дорожных плащей – мы отправляемся в мир волшебного искусства сисадмина Linux. **Ник Вейч** раскрывает книгу заклинаний…



Многим из нас для счастья вполне достаточно включить компьютер да и работать себе. И все бы отлично; ну, а вдруг компьютер заартачится? Или — вы знаете, что желаемое возможно, но не уверены в умении это реализовать? Лишние знания бывают крайне опасны, но и крайне полезны, если вы пытаетесь заставить свою Linux-машину плясать под вашу дудку.

Наши ожидания от компьютеров со временем изменяются, и если ваш дом похож на наш, он нашпигован хитрыми сетевыми устройствами, там явно не один компьютер плюс масса алчных до электроэнергии ящиков, мигающих лампочками и соединяющих все это добро между собой. Управление такими системами представляется жутью, и, будучи линуксоидом, вы можете ощутить себя в изоляции. И зря! Вот вам хорошая новость: вы не одиноки. Есть масса информации вам в помощь – и в Internet, и на самом вашем компьютере (вы еще не забыли про все эти man-страницы, которые собирались прочесть?). А теперь к вам в руки вдобавок попала и книга-талмуд с массой ценных заклинаний, которые помогут вам добиться поставленных целей. Прочите ее – и узнаете, какое могучее волшебство таится в вашем Linux-ящике…

> «Как линуксоид, вы можете ощутить себя в изоляции. И зря!»



Перенести пакеты в новую инсталляцию

Зачем? Вы покалечили свою систему, и теперь ее нужно переустанавливать; или вы просто хотите добавить диск, и вам неохота искать все ваши приложения заново.

Как? В Ubuntu или любой другой системе с менеджером пакетов **dpkg** определение состава установленных пакетов – задача тривиальная. Команда **dpkg** имеет для этого специальную опцию:

```
dpkg --get-selections
```

Команда выведет список установленных в системе пакетов. Чтобы обеспечить переустановку, сначала сохраним этот список, командой

```
dpkg --get-selections > packages.txt
```

Сохраните текстовый файл в безопасном месте (или отправьте его себе по почте), а затем переустановите Linux, установите новый диск или что вы там собирались сделать. Скопируйте текстовый файл в локальный каталог и дайте следующие команды:

```
dpkg --set-selections < selections.txt
dselect update
apt-get dselect-upgrade
```

Приложение звездного уровня

Back in Time

Цель: Упрощение резервного копирования

Сайт: http://backintime.le-web.org

» Ничего особенного здесь нет; *Back In Time* просто создает резервные копии указанных вами каталогов, что в принципе делается за пять минут из командной строки. Но вот в чем истинное преимущество этой программы: она настолько упрощает резервное копирование, что вы действительно станете его выполнять; а уж это всем магиям магия.

› **Половина битвы с резервным копированием будет выиграна, если вы просто начнете этим заниматься.**

Немного поколдуем

Вам нужно следить за своим хозяйством? Поможет команда **watch**. Она просто очень часто запускает другую команду, обновляя отображение ее вывода и, при необходимости, выделяя изменения. Например, чтобы наблюдать список файлов, скомандуйте:

```
watch -d ls -l
```

Советы по работе с оболочкой

1 Перенаправляйте вывод

Оболочка использует двойные угловые скобки для перенаправления стандартного вывода, и вы можете сохранить его в файл. Например, вот как можно записать отчет о занятости памяти:

```
free >> ram.txt
```

2 Фильтруйте файлы в grep

Можно перенаправлять вывод одной команды на вход другой, с помощью символа канала | связывая их в одну строку. Обычно это полезно для фильтрации вывода или аналогичных целей, как показано в следующем примере:

```
rpm -qa | grep 'python' > python-packages
```

3 Сэкономьте на печатании

Не приходилось ли вам ввести команду и в итоге получить только сообщение о том, что для нее нужны права root? И как тут не разозлиться? Но набирать команду повторно незачем. Просто воспользуйтесь «историей» ввода команд оболочки, вот так:

```
sudo !!
```

4 Учитесь зацикливаться

Не тратьте свою жизнь и не утруждайте пальцы, вводя монотонные последовательности одинаковых команд. Вспомните об умении *Bash* работать с циклами. Вот эти чудеса в действии:

```
for i in 1 2 3 4 5
do
touch myfile.$i
done
```

5 Сэкономьте на печатании, ч. 2

Bash способен на раскрытие скобок и перечислит для вас значения, разделенные запятыми и заключенные в фигурные скобки. Так, **"{One, two}"** превратится в **"one two»**. С виду это не особо полезно, пока вам не захочется скопировать какой-нибудь глубоко вложенный файл, например:

```
cp /etc/X11/xorg.conf{,.bak}
```

6 Пользуйтесь поиском

Команда **find** часто мелькает в списках советов по работе с *Bash*, но она умеет не только искать по именам. Ключ **-mtime** ограничивает поиск файлами, которые были изменены в пределах указанного количества дней назад. Если ваш компьютер взломали, проверьте, не испорчены ли ваши файлы настройки, командой

```
sudo find /etc -mtime -1 -name "*conf*"
```

7 Берите уроки истории

Bash ведет историю команд, так почему бы ею не воспользоваться? Нажмите Ctrl-R и начните ввод команды. *Bash* автоматически допишет ее до соответствующей команды, которую вы вводили недавно.Для принятия этой подсказки нажмите Enter.



Управление сервисами

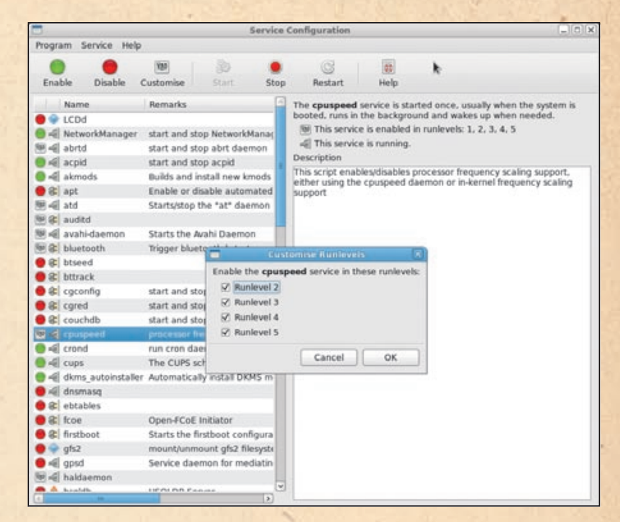
Зачем? Чтобы экономить системные ресурсы или добавить требуемые функции.

Как? При запуске вашего Linux-компьютера происходит инициализация большого количества программ. Многие из них настроены как сервисы – обычно это код, который незаметно работает в фоновом режиме, выполняя задачи вроде синхронизации системных часов или управления web-сервером. Многие из этих сервисов жизненно важны для безопасной работы вашего компьютера (например, *Cron*), но есть и избыточные – они подключаются при установке дистрибутива, потому что когда-нибудь могут понадобиться. Некоторые сервисы вообще могут бесить – например, сервис *ntp*, который синхронизирует ваши часы со службой точного времени через Internet. Это, возможно, и неплохо, если ваша машина постоянно сидит в сети, но годится не всем.

Практически во всех дистрибутивах имеется система управления сервисами. В Fedora это делается через опции System > Administration > Services. А Ubuntu имеет инструмент System > Administration > Startup Applications, хотя он настраивает не все сервисы в вашей системе. Но это не проблема: можно настроить систему и из командной строки.

«За кадром» эти системы запускают скрипты, содержащиеся в **/etc/init.d**, и иногда вы можете захотеть проделать то же самое.

Скрипты различаются друг с другом, поскольку поставляются пакетами, к которым относятся, но все они следуют общим правилам. Каждый предлагает возможность запуска или останова сервиса. Одни могут сообщать текущий статус, а другие умеют выполнять принудительный перезапуск – это бывает удобно



› **Fedora позволяет запускать сервисы однократно или постоянно, через System > Administration > Services.**

после выполнения правок. Например, ваши изменения, внесенные в конфигурацию *Apache*, не войдут в силу, пока вы не перезапустите сервер, вот так:

```
sudo /etc/init.d/httpd restart
```

Вы можете даже возжелать создать собственные скрипты для управления сервисами; а коли так, рекомендуем вам понравившееся читателям руководство по *Bash* из **LXF136**: там описаны основные принципы написания таких скриптов.

Немного поколдуем

Информационные страницы – это расширенные версии man-страниц, разбитые на тематические разделы и подразделы с удобной навигацией. Правда, удобна эта навигация, только если знать, на какие магические кнопки жать. К счастью, для них есть своего рода интерактивное руководство. Скомандуйте `info info` и нажмите клавишу h, чтобы начать процесс ознакомления.

Запуск задач из Anacron

Зачем? Чтобы ваш компьютер не был постоянно включен.

Как? *Anacron* – это напарник *Cron*, системного планировщика. *Cron* – отличный инструмент, гарантирующий, что необходимые задачи выполнятся. Но вот беда: задачи *Cron* назначаются на выполнение в конкретное время. Если ваш компьютер в этот момент будет выключен, *Cron* не сможет их запустить.

Выход – обратиться к *Anacron*. Когда система стартует, этот инструмент проверяет, нет ли невыполненных задач, и если они есть, пытается их запустить. Этот инструмент использует более простой метод планирования задач – ежедневных, еженедельных или ежемесячных – и при каждом запуске проверяет, были ли выполнены эти задачи в течение соответствующего временного интервала.

Итак, вы должны указать для *Anacron* временной интервал, задержку, идентификатор задачи и команду. Интервал – это количество дней между запусками задачи (например, для еженедельного это **7**), а задержка – это временная отсрочка (в минутах) с момента старта сервиса *Anacron*, чтобы все события разом не навалились на систему во время ее загрузки. Идентификатор задачи – это метка, необходимая для идентификации процесса в системных сообщениях или журналах, а команда – просто команда оболочки. Они перечисляются в файле **/etc/anacrontab** в виде списка, разделенного пробелами, который может выглядеть примерно так:

```
# /etc/anacrontab: файл настройки anacron
SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
# Максимальная случайная задержка относительно базового запаздывания заданий
RANDOM_DELAY=45
# Задания начнут запускаться только в указанные часы:
START_HOURS_RANGE=3-22
# Период в днях; задержка в минутах; индентификатор задания; команда
1         5    cron.daily      nice run-parts /etc/cron.daily
7         25   cron.weekly     nice run-parts /etc/cron.weekly
@monthly  45   cron.monthly    nice run-parts /etc/cron.monthly
7         55   anacron.backup  /usr/sbin/ mybackupscript
```

Приложение звездного уровня

Webmin

Цель: Управление всем подряд с безопасного расстояния

Сайт: http://webmin.com

» Если у вас есть удаленный Linux-компьютер, требующий заботы администратора, вам пригодится *Webmin*. Это простое приложение с web-интерфейсом упрощает поиск нужных элементов управления, какая бы версия Linux у вас ни была. Кроме того, программа пунктуально обновляется и отличается защищенностью.



› **Если надо — рулите своей Linux-машиной, хоть лежа в постели.**



Проверка средней нагрузки

Зачем? **Чтобы узнать, потянет ли ваш компьютер требуемые задачи.**

Как? Запустив команду **uptime** или заглянув в файл **/proc/loadavg**, вы увидите набор из трех чисел, указывающих среднюю нагрузку на ваш Linux-компьютер. И что же эти числа означают?

Во-первых, их три, потому что они получены усреднением по трем разным временны́м интервалам – одной, пяти и 15 минутам. Число указывает количество процессов, использующих CPU или ждущих своей очереди на его использование. Так, если ваша средняя нагрузка – 0,25, это значит, что процессор активно использовался в течение 25 % соответствующего временного интервала.

Для многоядерных машин ситуация чуть усложняется, поскольку число ядер не учитывается, и надо разделить на это число результат, показанный в файле **/proc/loadavg** или в выводе команды **uptime**. Например, средняя нагрузка 1 на четырехъядерную машину на самом деле будет означать то же самое – средняя нагрузка равна 0,25, то есть каждое ядро работает 25 % времени.

Средняя нагрузка – не самый точный показатель скорости CPU, так как процессы могут оказаться подвешены просто из-за медленного ввода/вывода; но дает представление о времени ожидания для пользователей. В идеале средняя нагрузка настольного компьютера должна быть меньше количества ядер. На сервере этот показатель может быть и превышен, но если средняя нагрузка постоянно высока, необходимо дальнейшее расследование.

Немного поколдуем

Оно вроде и очевидно, но стоит позаботиться о том, чтобы ваша аппаратура работала с максимальной скоростью. Например, большинство новых компьютеров, приводов NAS и др. имеют порты Gigabit Ethernet (10/100/1000), а для широкополосного доступа предоставляются маршрутизаторы только с портами Fast Ethernet (10/100), и соединение ваших устройств через такой маршрутизатор замедлит их работу. Купите маршрутизатор получше, или просто соедините более быстрые устройства через коммутатор Gigabit – скорости передачи данных между ними увеличатся раз в 10!

Добавление к sudoers через visudo

Зачем? **Чтобы добавить полномочий пользователям или процессам.**

Как? Команда **sudo** временно предоставляет привилегии root пользователям или процессам, чтобы те смогли выполнить задачу, разрешенную только root. Это особенно важно для систем типа Ubuntu: там нет пользователя root, и без команды **sudo** далеко не уйдешь.

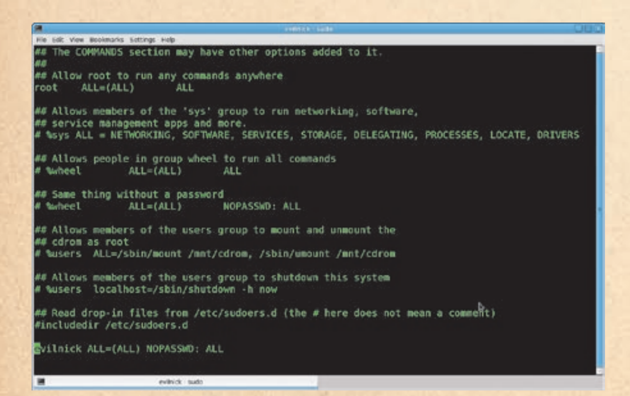
Сведения о том, какие права должны быть предоставлены и кому, хранятся в каталоге **/etc**, в файле под названием – вы удивитесь – **sudoers**. Его можно править в *Vi*, *Gedit* или любом другом редакторе, но рекомендуется применять *Visudo*. Данный текстовый редактор гарантирует, что ваш файл **sudoers** останется цел, пока вы будете его редактировать, поскольку запрещает одновременные правки и проверяет синтаксические ошибки. Это огромное преимущество в случае сбоев или просто человеческих ошибок, способных сделать файл **sudoers** нечитаемым, вынудив вас восстанавливать его из резервной копии.

> «Это огромное преимущество в случае сбоев или ошибок.»

Здесь действуют примерно те же правила, что и в *Vi*. Если вы с ними не знакомы – это означает, что перемещаться по файлу можно с помощью клавиш курсора (а также Page Up/Page Down), для перехода в режим ввода необходимо нажать клавишу **i**, после чего можно начинать вводить текст. Завершив ввод, наберите последовательность **Esc:wq <return>**. Если при редактировании файла вы допустите ошибку, часто бывает проще закрыть его без сохранения, а затем попытаться вызвать его еще раз. Для выхода из редактора введите комбинацию клавиш **Esc:q! <return>**.

Например, чтобы дать через **sudo** пользователю **evilnick** все права root без необходимости вводить его пароль – это опасно, но экономит время – добавьте

```
evilnick ALL=(ALL) NOPASSWD: ALL
```



› *Visudo*, конечно, дубоват, зато в нем не опасно редактировать файл sudoers; а его правила сходны с правилами *Vi*.

Движемся дальше

Посмотрев в начало файла, вы обнаружите, что там определены некие группы. По сути, это метки, применяемые к списку исполняемых файлов (с полными именами путей). При желании вы можете создать собственные группы, но уже готовые обычно определены очень хорошо.

Вместо отдельных пользователей можно также указать и группы, просто поместив символ **%** перед именем группы. Например, если вы хотите разрешить пользователям из группы **users** доступ к командам из группы **Networking**, которые обычно определяются здесь через **sudo**, и позволить пользователю **mike** загружать драйверы ядра через **modprobe**, добавьте такие строки:

```
%users ALL=NETWORKING
mike ALL=/sbin/modprobe
```

Заметьте, что первое **ALL** относится к хосту, так как файлы **sudoers** часто предоставляются в общий доступ через сеть, и при желании здесь можно указывать отдельные хосты.

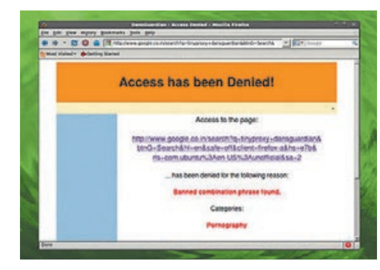
Приложение звездного уровня

Dansguardian

Цель: Безопасный семейный серфинг

Сайт: http://dansguardian.org

» *Dansguardian* – это прокси с web-фильтрацией, который не просто полагается на наскоро созданные черные списки, но и фильтрует содержимое web-страниц, определяя, следует ли его отображать. Естественно, это нельзя считать полноценной заменой контроля за сайтами, которые посещают ваши дети, но может предотвратить их случайные попадания в грязные закоулки сети.



› *Dansguardian* защитит ваш компьютер от всякой дряни.



Копирование файловых систем целиком

Зачем? Чтобы клонировать свой ПК или сделать полную резервную копию.

Как? Существует далеко не один рецепт, как начинить индейку; то же касается создания копий системы Linux. Возможно, простейший из них – применить команду **dd**, обычно используемую для поблочного копирования больших файлов. Однако, раз уж в Linux практически все является файлами, ту же команду можно применить и для копирования всего диска. Содержимое одного жесткого диска IDE копируется на другой при помощи команды

```
dd if=/dev/hda of=/dev/hdb conv=noerror,sync
```

Дополнительные опции этой команды позволяют пропускать ошибки чтения (что удобно при копировании порченого диска) и принудительно синхронизировать копии.

> «Работать-то dd работает, но по ряду причин стоит ее избегать.»

Работать-то она работает, но по ряду причин стоит ее избегать. Во-первых, это процесс весьма долгий – команда **dd** копирует каждый блок диска, включая и неиспользуемое пространство. Другая проблема – в том, что ваш второй жесткий диск должен быть по возможности идентичен с первым. При использовании данного метода копируется вся информация, и если второй жесткий диск имеет больший размер, окажется, что вы зря теряете дисковое пространство.

Еще одна потенциальная проблема – при таком копировании установки Linux скопируется и содержимое виртуальных файловых систем, например, **/proc**, что тоже чревато неприятностями.

Если вам нужна копия файловой системы диска, лучше возьмите команду **rsync**:

```
rsync -av /source /target
```

В особом случае копирования корневой файловой системы, где содержится загружаемая копия Linux, вы, возможно, захотите исключить из копирования часть файлов – например, те, что система генерирует при загрузке. Для этого сгенерируйте список файлов и назовите его **blacklist.lst**:

```
# файлы для включения
+ /dev/console
+ /dev/initctl
+ /dev/null
+ /dev/zero
# файлы для исключения
- /dev/*
- /proc/*
- /sys/*
- /tmp/*
```

Некоторые записи **dev** включены явно, поскольку они не воссоздаются повторно, как и каталог **/proc** и некоторые другие каталоги: содержимое каталога регенерируется, а сам каталог – нет. Теперь запустите команду **rsync** со следующими параметрами:

```
rsync -av --delete-excluded --exclude-from=blacklist.lst / /
destination
```

Учтите, что это можно сделать и через сеть, но тогда надо сперва должным образом настроить **rsync**.

Немного покодуем

Начиная с серии ядер Linux 2.6, параметр 'swappiness' стал управлять тем, насколько агрессивно ядро пытается освободить память, вытесняя ее содержимое в пространство подкачки. Значение этого параметра можно найти в файловой системе **/proc**:

```
cat /proc/sys/vm/swappiness
```

Чем выше значение swappiness, тем больше данных вытесняется в раздел подкачки. Это поведение можно изменить, записав в **proc** другое значение, но слишком завышать его не стоит – если вы часто переключаетесь между приложениями, ваша система станет менее отзывчивой.

Проверка проблем с помощью dmesg

Зачем? Простой способ выявить проблемы с драйверами нужен всем.

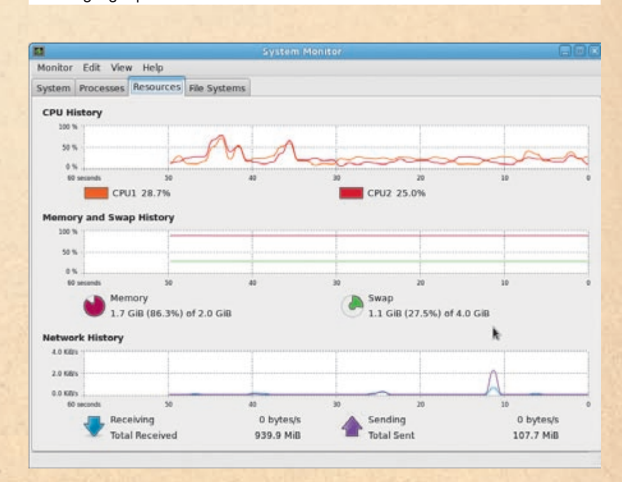
Как? Linux и ряд других Unix-подобных операционных систем включают в свой состав очень полезную команду под названием **dmesg**: она отображает все системные сообщения, имеющие отношение к проблемам, в своем журнале. Это ценная информация, в частности, о драйверах устройств, а также о том, загружены ли они корректно, плюс еще набор полезных сведений. Для просмотра журнала просто запустите команду **dmesg** из оболочки.

Запуск **dmesg** сразу же после добавления аппаратного устройства сообщит вам, загружен ли драйвер устройства и работает ли он правильно, или о том, где были выделены ресурсы; но иногда бывает нужно просмотреть журнал. Здесь неплохой идеей будет перенаправить вывод команды **dmesg** команде **less**:

```
dmesg | lessa
```

Если вы ищете что-то конкретное – например, устройство SCSI или USB – всегда можно просеять журнал через **grep** и перенаправить его в **less**:

```
dmesg | grep 'usb' | less
```



› Примерную оценку использования подкачки даст либо *Gnome System Monitor*, либо команда free.

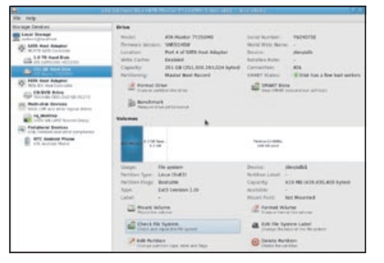
Приложение звездного уровня

Palimpsest

Цель: Высший пилотаж управления дисками

Сайт: http://library.gnome.org/users/palimpsest/2.32/intro.html.en

» *Gnome Disk Utility* или *Palimpsest* (название пергамента, очищенного для повторного использования) — великолепный инструмент для управления дисками, хотя лишь немногие вообще знают о его существовании. Помимо обычных монтирования и форматирования, вы можете редактировать разделы и запускать диагностические средства на подключенных носителях. И это средство устанавливается как часть Gnome, так что употребляйте!



› Утилиты диагностики диска просты и помогают избежать потери данных. Почему бы их не запустить?



Управление разделом подкачки

Зачем? **Чтобы ускорить вашу систему или приспособить ее к обстоятельствам.**

Как? Раздел подкачки очень важен для Linux-систем, или, по крайней мере, бывает очень важным. Он позволяет ядру отложить в сторонку информацию, которая не является необходимой в данный конкретный момент, и освободить ОЗУ системы под работающие процессы. Старое эмпирическое правило гласит, что объем этого раздела должен быть не меньше объема физического ОЗУ, а желательно – вдвое больше, но это правило больше не покрывает все варианты.

> «Интересно, что ядро чередует доступ к разделам подкачки.»

Во-первых, в наши дни вы, скорее всего, богаты физическим ОЗУ, поскольку цены на них стали вменяемыми и при желании вполне можно всадить на материнскую плату 8, а то и 16 ГБ. Поэтому слишком большой раздел подкачки уже ни к чему, но системные администраторы считают идеальным объем около 2 ГБ.

Чтобы узнать фактически используемый объем раздела подкачки, запустите либо команду **free**, либо команду **swapon**, например:

```
swapon -s
```

Выведется полный объем раздела подкачки в байтах.

Однако, возможно, вам потребуется более эффективно управлять разделом подкачки, если вы запускаете много экземпляров параллельно работающих сервисов (например, баз данных или web-серверов).

Увеличение раздела подкачки

Создав для подкачки дополнительный раздел, можно добавить его к текущему (считая, что новый раздел подкачки – **sdb2**) командами

```
mkswap /dev/sdb2
swapon /dev/sdb2
echo '/dev/sdb2 swap swap sw 0 0 ' >> /etc/fstab
```

Последняя строка добавляет новый раздел подкачки на постоянной основе, и он будет монтироваться при загрузке.

Интересно, что ядру хватает ума на чередование доступа к разделам подкачки, и вы можете серьезно повысить производительность, предусмотрев разделы подкачки на всех быстрых жестких дисках, имеющихся у вас в системе.

Кроме того, можно использовать для подкачки файл. Хотя обычно это не лучший способ решения проблемы, он помогает в ситуациях, когда требуется временно удалить диск из системы или обеспечить себе дополнительное место для подкачки (например, при визуализации огромной и детальной 3D-сцены).

Чтобы обзавестись файлом подкачки, можно создать пустой файл командой **dd**, а затем его смонтировать:

```
dd if=/dev/zero of=/var/swapfile bs=1024 count=2097152
mkswap /var/swapfile
swapon /var/swapfile.
```

Наконец, заметим, что создание раздела подкачки на медленных устройствах – например, USB-брелках – идея отнюдь не здравая. Ядро, конечно, съест и такое, если добавить его к системе, но скорость USB 2.0, около 400 Mbps, существенно снизит производительность. А вот быстрый твердотельный диск – идеальное место.

Немного поколдуем

При вводе команде **sudo** неверного пароля появится зловещее сообщение о замеченном инциденте – угрозе безопасности. Но кто его получит? Это зависит от настройки вашей системы. Можно указать почтовый адрес при сборке пакета **sudo** или переопределить его в файле **sudoers**. Обычно это адрес почты админа. А на самом деле кто? См. http://xkcd.com/838.

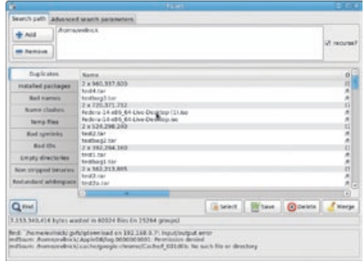
Приложение звездного уровня

FSlint

Цель: Возврат потраченного пространства

Сайт: www.pixelbeat.org/fslint

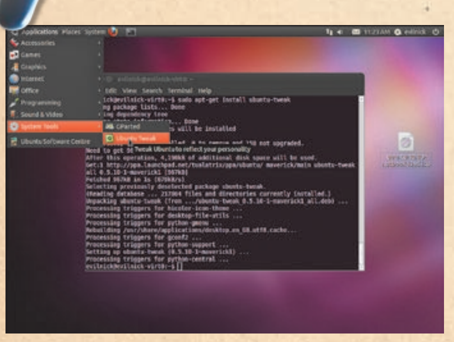
Изрядная часть вашего дискового пространства тратится без толку. По сути, оно расходуется вхолостую, медленно деградирует и хранит избыточные копии информации. Это только одна из областей, где вам может помочь *FSlint*. Программа обыщет все области вашего диска и очистит их для вас, выкинув ненужный хлам.

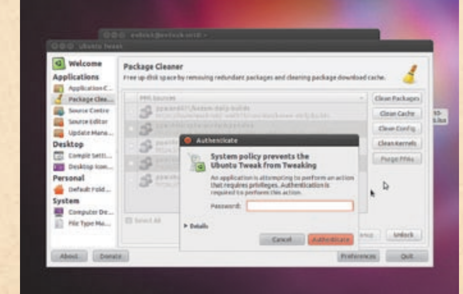


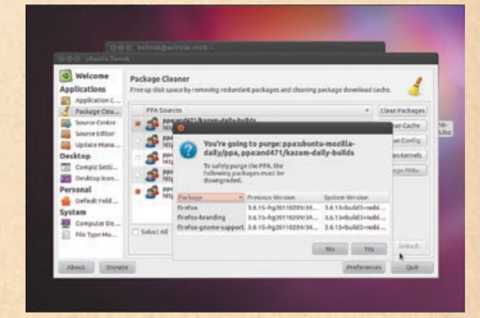
› Эй, *FSlint*. Как отработаешь — может, заодно и полы подметешь?



Шаг за шагом: Откат обновлений PPA







1 Ставим Ubuntu Tweak

Ubuntu Tweak – отличный инструмент для управления пакетами Ubuntu. Он устанавливается так:

```
sudo add-apt-repository ppa:tualatrix/ppa
sudo apt-get update
sudo apt-get install ubuntu-tweak
```

Теперь инструмент появится в меню System Tools.

2 Чистим PPA

Выберите опцию Package Cleaner в левом меню, затем опцию Purge PPA в правом нижнем углу главной панели. На данном этапе, вероятно, все будет блокировано. Щелкните по кнопке, снимающей блокировку, и введите пароль, чтобы авторизоваться для внесения изменений.

3 Шагаем назад

В нашем случае выбрано удаление *Firefox* и ряда других программ. На самом деле, мы не хотим остаться без *Firefox* или удалить все, что имеет к нему отношение. *Ubuntu Tweak* понимает это и дает нам возможность снизить версию пакета до последних репозиториев Ubuntu. Дело сделано!



Ручное добавление и удаление драйверов

Зачем? Потому что драйверы по умолчанию годятся не всегда.

Как? Преимущество современных дистрибутивов Linux – отличная поддержка драйверов. Серьезно, взять произвольное устройство и добиться его работы с Linux намного проще, чем с Windows или Mac OS X, потому что Linux добавляет функции драйверов в ядро посредством модулей. Код модулей открыт, и, за исключением особых обстоятельств, вам ничто не мешает обновить код старого драйвера, чтобы он поладил с новейшей версией ядра.

> «Lsmod покажет модули, загруженные в данный момент.»

Вы можете выяснить, какие модули работают в вашей системе, командой **lsmod**. Она покажет вам загруженные на данный момент модули, а также список модулей, от которых те зависят. Например, модуль **kvm_intel** зависит от наличия модуля **kvm**, а модуль **snd** – от модуля **sndcore**.

При необходимости можно удалить драйверы из работающей системы с помощью **rmmod** и вновь загрузить их с помощью **modprobe**. Например, чтобы отключить и вновь включить джойстик, скомандуйте

```
sudo rmmod joydev
sudo modprobe joydev
```

Знайте также, что иногда при загрузке модулей доступны опции. Драйверы web-камер часто имеют необязательные параметры, позволяющие отключить инфракрасные светодиоды или изменить размер кадра либо частоту смены кадров. Вы можете прочесть документацию о модуле либо применить команду **modinfo**. Как правило, список доступных параметров приводится в конце, например:

```
$ modinfo gspca_ov519

filename:    /lib/modules/2.6.35.10-74.fc14.x86_64/kernel/
drivers/media/video/gspca/gspca_ov519.ko
license:     GPL
description: OV519 USB Camera Driver
...
alias:       usb:v041Ep4003d*dc*dsc*dp*ic*isc*ip*
depends:     gspca_main
vermagic:    2.6.35.10-74.fc14.x86_64 SMP mod_unload
parm:        frame_rate:Frame rate (5, 10, 15, 20 or 30 fps) (int)
```

Отсюда видно, что при загрузке модуля можно указать частоту смены кадров вручную. Конечно, неплохо иметь такую возможность; ну, а если вы хотите задать эти опции по умолчанию?

```
sudo rmmod gspca_ov519
sudo modprobe gspca_ov519 frame_rate=20
```

Для этой цели предусмотрен специальный файл – обычно он называется **/etc/modules.d/options.conf**. (Если он не существует, создайте его.) Добавьте в него такую строку:

```
options gspca_ov519 frame_rate=20
```

Она гарантирует, что данные опции будут браться при каждой загрузке модуля, если вы не отмените их вручную.

Кстати, в том же каталоге обычно имеется и файл **blacklist.conf** – он пригодится, если понадобится отключить проблемное оборудование или драйверы, которые норовят предъявить права на аппаратные устройства первыми.

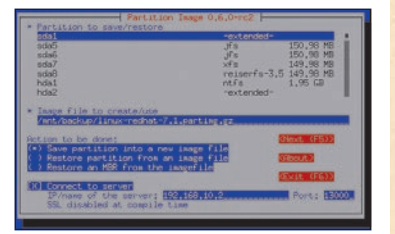
Приложение звездного уровня

Partimage

Цель: Клонирование ваших дисков

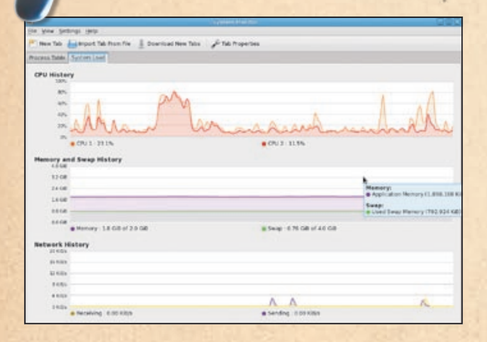
Сайт: http://www.partimage.org/Main_Page

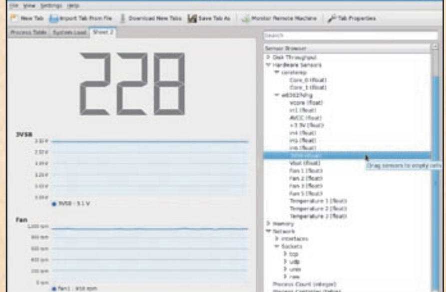
» *Partimage* — инструмент клонирования дисков, способный создавать образы разделов или устройств и даже их резервные копии через сеть. Лежащее в его основе ПО использует интерфейс Curses, и его требования минимальны; но вы можете также рассмотреть одну из опций дисков аварийного восстановления, для простоты использования включающих *Partimage* — например, *Clonezilla* (см. http://clonezilla.org).

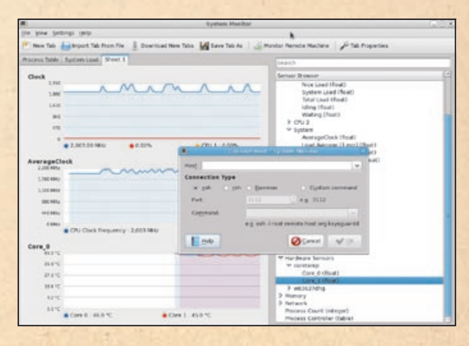


› **Вы увидели привидение? Да бросьте, это просто *Partimage* копирует ваши диски.**

Шаг за шагом: Мониторинг системы с Ksysguard







1 Запуск Ksysguard

Ksysguard – часть базовой инсталляции KDE, и если у вас есть KDE, то *Ksysguard* у вас тоже есть. Запустив эту программу, вы получите стандартное окно мониторинга ресурсов. Оно очень похоже на эквивалент из Gnome, но дает намного больше возможностей!

2 Выбор новых опций

На панели инструментов выберите New Tab, и создастся новая пустая вкладка. Из расположенного справа дерева теперь можно перетаскивать мышью свойства, за которыми вы решили наблюдать. *Ksysguard* также отображает аппаратные мониторы вашего CPU и материнской платы.

3 Следите за всем

На один и тот же график можно перетащить множество показателей и мониторить, например, все операции доступа к диску. Если у вас больше одной системы, можно наблюдать за всеми работающими системами в одном окне. Для установки соединения щелкните по кнопке Monitor Remote Machine.

NAS: Сделать самим или купить?

Возможно, вас посещала идея превратить свой старый, ни на что другое не годный ПК или ноутбук в сетевое устройство, чтобы хранить там резервные копии, слать через него поток медиаданных и прочее. Да, это можно; но не факт, что все будет так замечательно, как вы воображали. ПК, превращенный в NAS, хорош тем, что может работать как полноценная версия Linux или настоящий сервер. Однако большинство ПК не рассчитано на круглосуточную работу, и даже если их к этому принудить, очередной счет за электроэнергию вас скорее всего неприятно удивит. Современные NAS-устройства, например, Qnap 219 или Excito B3, умеют интеллектуально управлять электропитанием и потребляют около 20 Вт (а не 200), а также заранее настроены на выполнение функций устройств сетевого хранения данных.



› Выделенное NAS-устройство поможет вам оптимизировать расходы на оплату электроэнергии.

Защита паролем при загрузке

Зачем? Чтобы допускать к определенным инсталляциям и вообще к загрузке компьютера только доверенных пользователей!

Как? Большинство основных дистрибутивов переходят или уже перешли на загрузчик *Grub 2* (заметное исключение – Fedora). Это позволяет установить аутентификацию пользователей для пунктов загрузочного меню и как минимум отчасти защитить систему(ы) от несанкционированного доступа: например, разрешить только некоторым пользователям загружать конкретные ядра.

Во время загрузки меню отображается как обычно, но когда кто-то попытается вызвать защищенный пункт загрузочного меню, ему предложат ввести имя пользователя и пароль. Выполнить эту настройку не так уж просто, но мы рискнем. Первый порт вызова – заголовочный файл. Он находится в **/etc/grub.d/** и называется **00-header** или похоже. Откройте его для редактирования в любимом текстовом редакторе и добавьте в его конец текст

```
cat << EOF
set superusers="myname"
password myname 1234
EOF
```

Он устанавливает пользователя (**myname**) и его пароль (**1234**). Имейте в виду, что если вы здесь что-то напутаете, обвинять будет некого, кроме себя, поэтому позаботьтесь, чтобы в меню был пункт, из которого вы сможете загрузиться и исправить дело, если что-то пойдет не так.

Теперь отредактируйте файл **10-linux** из того же каталога. Вам нужны записи, начинающиеся с

```
menuentry "$1" {
```

или с

```
printf "menuentry \"${title}\" --class linux {
```

Добавьте в эти строки переключатель **--users** и укажите имя пользователя сразу же после открывающей фигурной скобки или элемента меню, например:

```
menuentry "$1" --users myname {
```

или

```
printf "menuentry \"${title}\" --class linux --users myname {
```

Сохраните файл и обновите *Grub* командой:

```
sudo update-grub
```

Теперь вы готовы перезагрузить компьютер и протестировать новое загрузочное меню, защищенное паролем. Можете даже подшутить над любителями влезать в вашу машину: не говорите им, что защитили загрузочное меню паролем, и дождитесь, чтобы они прибежали к вам с вопросами. Вот, кстати, и причина не делать пароли слишком простыми для угадывания – например, просто 'password' или ваш день рождения.

Немного поколдуем

Регистрируетесь на удаленных машинах? Люди при этом склонны забывать, на каком компьютере находятся. Поэтому не пожалейте времени на подобающую настройку приглашения командной строки, чтобы отличать компьютеры, на которых вы регистрируетесь, друг от друга. Попробуйте добавить следующий текст в файл **.bashrc** в вашем каталоге удаленного пользователя:

```
export PS1="\
[\e[41;1m\]\u@\
[\e[32;1m\]\H> \
[\e[0m\]"
```

Остановите SSH-атаки

Зачем? Когда ваш компьютер доступен через Internet, каждая свинья рвется получить к нему доступ.

Как? Открыв доступ к своей машине через Internet – скажем, вы хотите работать на своем домашнем компьютере, будучи в отъезде, или держите web-сервер – вы поразитесь количеству зондирующих атак злоумышленников. Проверки ради, за 60 минут онлайн мы насчитали 43 потенциальных взломщика, каждый из которых неоднократно пытался войти на SSH-сервер.

Есть несколько простых вещей, которые можно сделать, чтобы отражать атаки. Во-первых, проставьте другой порт в файле настройки SSH взамен стандартного 22. Это малость осложняет, поскольку потом нужно будет запомнить номер порта и вручную вводить его при каждой регистрации. Во-вторых, можно блокировать регистрацию с паролем. Но тогда придется установить ключи SSH, чтобы иметь возможность регистрироваться, причем доступ будет выдаваться только известным компьютерам.

Еще один трюк – включать SSH только тогда, когда это действительно требуется. Мы однажды пробовали метод секретного кодового слова, которое отправляли на специальный адрес электронной почты. Кодовое слово перехватывал скрипт на сервере и на час включал SSH. Это было очень хитро, но канительно. Но каким бы вариантом вы ни воспользовались, все равно понадобится развернуть скрипт *fail2ban*.

Он будет читать ваши журналы и определять, кто предпринял неудавшиеся попытки регистрации. После некого числа неудачных попыток регистрации – скажем, шести за 10 минут – скрипт активизирует *iptables*, чтобы блокировать соединение с вашим компьютером с этого хоста на заданный промежуток времени. По умолчанию, это 10 минут – вроде и недолго, но большинство таких атак осуществляется скриптами, которые через несколько неудачных попыток прекращают атаку и переключаются на другую мишень.

Скрипт *fail2ban* вы найдете на сайте sourceforge: www.fail2ban.org/wiki/index.php/Main_Page. Внимательно прочтите документацию по настройке этого сервиса и не отделывайтесь значениями по умолчанию. В порядке бонуса, скрипт дает возможность наблюдать за попытками взлома и других сервисов. LXF



Обратная связь

Когда вы следите за системой Linux, обычное дело – записывать, что именно вам помогло. Если у вас есть советы, которыми вы хотели бы поделиться, мы с удовольствием их выслушаем, а лучшие напечатаем в одном из будущих номеров.

19 перемен к лучшему для KDE

Последние релизы KDE вызвали неоднозначную реакцию. Улучшения есть, но продукт еще далек от идеала. **Нейл Ботвик** ныряет в первичный плазмоидный супчик, чтобы наловить предложений по усовершенствованию.

01 Хватит добавлять столько функций

KDE нуждается в стабильности, в обоих смыслах этого слова. Хватит перетасовок – лучше будет обеспечить, чтобы работа с KDE снова стала приятной. Стабильность, в том смысле, что среда больше не падает, сейчас улучшилась, но приноравливаться к ней после каждого обновления довольно утомительно. Это напоминает супермаркет, в котором кофейня постоянно переезжает, и в поисках живительной жидкости вы обшариваете всю торговую площадь.

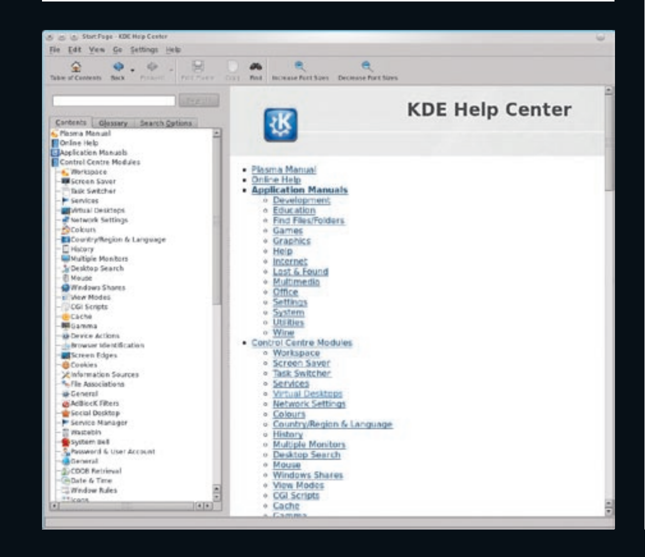
02 Упростить поиск опций настройки

Панель KDE System Settings – это огромный монстр, состоящий из множества отдельных модулей, отчего найти опцию для внесения требующегося изменения намного сложнее. Поле Search серьезно помогает в этом, но оно уж слишком хорошо спрятано. И нужна альтернатива – скажем, встроенный в программы FAQ для вопросов типа «Как мне настроить...»

03 Названия программ

Нужна ли нам столь навязчивая помощь в узнавании программ KDE? Неужто мы с этим не справимся, если Каждое Криложение не будет обозвано Какой-то Корявой Кличкой на букву K? В другой рабочей среде Linux все-таки не у всех программ имя начинается с G... Или вы такое встречали?

> «Нужна ли нам столь навязчивая помощь в узнавании KDE?»



› *Help Centre* — неплохой справочник, но где полезные обучающие материалы или даже видео?

04 Приятный вид программ *GTK* по умолчанию

То, что программы *GTK* плохо выглядят в KDE (а приложения *Qt* – в Gnome), считается прописной истиной, но это не обязательно должно быть так. Существуют сторонние опции подстройки темы *GTK* под текущую тему *Qt*, и их следует встроить в KDE. Бессмысленно делать вид, будто пользователи KDE никогда не работают с программами *GTK*, даже если это всего лишь *Firefox* для доступа к сайтам, с которыми не справляется *Konqueror*.

05 Про имя

В имени «KDE» ничего плохого нет. Но оно слишком напоминает популярную, но в корне иную среду рабочего стола, бывшую в ходу пару лет назад. Она называлась KDE 3.5 или часто просто KDE, и делала все, что требовалось ее пользователям, понятным им способом. Дать то же имя абсолютно другому продукту – не самый верный ход.

06 Больше руководств

KDE Help Centre – хороший источник справочной информации, но KDE – сложный комплект ПО, способный выполнять множество различных задач. Сейчас в документации KDE отчаянно не хватает учебников и пошаговых инструкций. Переходный период был бы менее мучительным, если бы появилась серия руководств по выполнению типовых задач KDE 3 в среде KDE 4.

07 Видеодокументация

Сервис YouTube показал нам пользу и информативность видеокурсов (и не только это, но о другом мы здесь говорить не будем). Набор вводных видео, поясняющий, что есть что – и новым пользователям, и тем, кто переходит с KDE 3 – мог бы здорово улучшить репутацию KDE.

08 Опять про имя

KDE был настольной средой, каковой термин означал (Констатировал?) полностью интегрированный набор ПО. Теперь это просто коллекция программ, компоненты которой связаны между собой гораздо слабее. KDE – единое целое, и надо этим гордиться!

09 Многоуровневые опции настройки

Раньше между стандартными и расширенными настройками существовало разделение, которого, увы, больше

нет. Мы все же считаем, что глобальная опция, позволяющая показать или скрыть дополнительные настройки, но изменяемая нажатием кнопки внутри каждой секции, могла бы разгрузить экран. Возможно, имеет смысл ввести профили, чтобы, например, в профиле для одиночного настольного компьютера не болтались управление сетью или настройки батареи и сенсорной площадки.

10 Управление сетью

Где в KDE эквивалент *Network Manager*? Как должны мобильные пользователи KDE переключаться между различными беспроводными, 3G- и VPN-соединениями? Это – основное требование, предъявляемое к современным ноутбукам, а KDE до сих пор не предлагает удобного способа переключения между сетями без необходимости их предварительной ручной настройки.

11 Снизить потребление ресурсов

Несомненно, KDE 4 съедает много системных ресурсов. Мы можем запустить эту среду на многоядерном процессоре или на компьютере, имеющем более 4 ГБ ОЗУ, но ресурсы нам нужны для выполнения наших программ, а не для того, чтобы окна порхали по экрану. Красивый интерфейс хорош для пускания пыли в глаза друзьям из мира Windows, но было бы неплохо иметь более поджарую и шуструю рабочую среду, хотя и со всеми функциями KDE.

> «В KDE 4 много здравых идей, но где былая радость?»

12 Меньше дешевого лоска

Популярность дистрибутива Ubuntu на основе Gnome показывает, что люди не обязательно стремятся к разным красотам или даже к приятной цветовой гамме. Десятки пестрых плазмоидов, часто дублирующих функции друг друга – это не тот способ привлечения новых пользователей, в котором нуждается KDE.

13 Улучшить управление питанием

В KDE 4.5 были удалены некоторые опции по управлению питанием, в частности, относившиеся к масштабированию CPU. Код управления по-прежнему на своем месте, но GUI к нему ликвидирован. KDE всегда считался хорошим выбором для любителей повозиться с тонкой настройкой, поэтому пора бы перестать обращаться с ними, как с пользователями Gnome.

14 Простой исходный интерфейс

Gnome – интерфейс простой и опрятный; он идеален для новых пользователей. В KDE есть все необходимое опытным пользователям, но – и это весомое «но» – люди не рвутся менять рабочие столы, в которых им удобно. А это значит, что если новичок дорастет в Gnome до того уровня, когда ему, возможно, понравится KDE, будет уже поздно пытаться привлечь его внимание.

15 Показать простоту в использовании

О сложности KDE говорят много. Конечно, данная среда изобилует тонкими опциями, но сложной ее делает не это. Набор предопределенных конфигураций (профилей, если хотите), способствующий быстрой настройке среды для разных пользователей в разных ситуациях, помог бы развеять миф «KDE – штука сложная».

› Опций много, а нужной нет. Не реорганизовать ли System Settings? Во-во, опять начинается…

16 *Konqueror* требует улучшения

В эру KDE3 мы использовали *Konqueror* как web-браузер, менеджер файлов и даже средство доступа к удаленным файловым системам. KIO-slaves, которые делают это возможным, сегодня хуже не стали. Но *Konqueror* в качестве web-браузера слишком медлителен по сравнению с *Chrome* и ему подобными, и справляется не со всеми сайтами.

17 Отвоевать потерянных пользователей

Ранние релизы KDE 4 причинили немало вреда. Многие пользователи выбрали альтернативы: одни – из-за проблем со стабильностью, другие – потому что не поняли и не приняли новых концепций KDE. В глубине души многие из этих потерянных пользователей все еще любят KDE, или, по крайней мере, KDE 3; их-то и нужно попытаться вернуть. Это не коммерческое предприятие, и рост числа пользователей не означает прямых прибылей, но все же чем больше сообщество, тем мощнее продукт.

18 Понять, что удовольствие главнее пользы

Несомненно, в KDE 4 много здравых идей, но где былая радость от работы с ним? Когда работа становится рутиной, недалеко и до перехода на альтернативу. Нам не нужны клоуны, скачущие по экрану, или что-то еще в этом роде, но дайте пользователям возможность работать так, как они хотят. Ведь это всегда было одной из самых сильных сторон KDE.

19 Никакого коричневого. Никогда

Хотя для некоторых сред он, похоже, неплохо работает, пожалуйста, не думайте, что перекрашивание всего подряд в этот скучный коричневый цвет добудет вам популярность. Это не так. LXF

Оговорка

Внимание: эта статья может спровоцировать острый приступ злости и потому не рекомендуется людям с сердечно-сосудистыми заболеваниями, беременным женщинам и детям младше пяти. А остальные... прежде чем ошпаривать нас своей яростью, расслабьтесь. На самом деле мы любим KDE. Нет, правда – вы только посмотрите наш материал о Plasma. По сути, мы относимся ко KDE как к младшему братишке, которого у нас никогда не было, и если мы его чуточку подразним, ничего худого с ним не случится. Если вы все еще гневаетесь, мы откомандируем нашего обаяшку Майка вас приласкать и утешить...

Мы посмотрим, что внутри AR.Drone

Это диво-квадрикоптер – самая компактная и простейшая из летающих копий Linux. Как понял **Грэм Уэлдон**, с ним еще и прикольно.



AR.Drone от Parrot добился колоссальных успехов с момента выпуска в сентябре 2010 года, и постоянно поражает игровые сообщества и сообщества разработчиков. Более того, покопавшись в нем, мы обнаружили, что AR.Drone работает под Linux. Здесь мы рассмотрим, что делает этот гаджет.

Подключить его к вашему компьютеру очень легко, потому что это устройство Wi-Fi. Как таковое, он выглядит специальной сетью, к которой затем можно присоединиться с любого компьютера, имеющего беспроводную сетевую карту. Отметим, однако, что если вы уже соединили его с мобильным устройством, IP-адрес не будет назначаться при подключении к сети, и к нему невозможно будет подсоединиться. Чтобы обойти эту проблему, запустите мобильное устройство и отключите функцию спаривания в настройках приложения *Free Flight*, или просто нажмите кнопку Разорвать пару [Unpair] на устройстве.

> «Устройство Wi-Fi выглядит как специальная сеть.»

Итак, что мы можем почерпнуть из этой игрушки, не разбирая ее на части? Так как мы в его сети и знаем его IP-адрес, сканирование портов будет хорошей отправной точкой. Выполнив это, мы узнаем, какие порты открыты и доступны, а также, вероятно, поймем, какой доступ мы получим для подключения к устройству.

Запуск *nmap* показал два интересующих нас порта:

```
21/tcp open ftp
23/tcp open telnet
```

Попытавшись установить подключение к свободному порту telnet, мы попадем в *Bash* без необходимости ввода пароля. Информация по входу показывает, что нам удалось подключиться к установке Linux с *BusyBox* и что есть доступный номер версии, поэтому мы можем проверить интимные подробности и вопросы безопасности, если это необходимо.

Дальнейшее рассмотрение показывает, что нас на самом деле непринужденно сбросили под root без пароля, и мы получили полный доступ к устройству.

```
# echo $USER
root
```

Как-то уж слишком легко. Леденею при мысли, что кто-то по соседству может подсоединиться к сети с нетбука и поубивать процессы, пока я буду парить на квадрикоптере в небесах.

Обзор оборудования

Давайте взглянем на информацию по процессору и памяти:

```
# cat /proc/cpuinfo
Processor: ARM926EJ-S rev 5 (v5l)
BogoMIPS:  233.47
```



```
Features: swp half thumb fastmult edsp java
CPU implementer: 0x41
CPU architecture: 5TEJ
CPU variant: 0x0
CPU part: 0x926
CPU revision: 5
Cache type: write-back
Cache clean: cp15 c7 ops
Cache lockdown: format C
Cache format: Harvard
I size: 32768
I assoc: 4
I line length: 32
I sets: 256
D size: 16384
D assoc: 4
D line length: 32
D sets: 128
Hardware: Mykonos Parrot platform
Revision: 0904
Serial: 0000000000000000
```

А также информация относительно памяти:

```
# cat /proc/meminfo
MemTotal: 126072 kB
MemFree: 105652 kB
Buffers: 0 kB
Cached: 3604 kB
```

Здесь 128 МБ оперативной памяти на борту – немного больше, чем вы ожидаете найти на столь маленькой штучке. Памяти и вычислительной мощности хватает для достойного встроенного устройства, и мы знаем, что на нем как минимум работают демоны *FTP* и *telnet*. Пора просмотреть полный список процессов и увидеть, что еще есть внутри.

Подберемся поближе

Взглянув на список запущенных процессов, мы найдем не так уж много полезных для нас, но среди них есть демон *FTP*, а также *Telnet*, DHCP-сервер и syslogger. Кроме того, есть два интересных процесса, которые выглядят специфичными для этого устройства, и, глядя на полный список всех запущенных процессов, можно заключить, что один из них отвечает за обработку всех элементов управления и вывода видео обратно на ваше мобильное устройство. Соответствующая информация здесь приводится ниже:

```
960 root    2736 S   /bin/sh /bin/check_update.sh
961 root   11824 S   /bin/program.elf
```

С виду этот процесс **program.elf** похож на хостинг основного кода устройства. Здесь недоступен исходный код, и он, вероятно, был прикомпилирован к материнской плате, чтобы закрыть исходный код от пользователей.

Процесс **check_update.sh** выглядит доступным для чтения – почитаем его, запустив **vi** или **cat**:

```
# cat /bin/check_update.sh
```

Похоже, что он контролирует каталог прошивки в поисках новых версий для обновления. Итак, остался только один процесс – **program.elf**. Он, вероятно, отвечает за всю коммуникацию и взаимодействие с мобильными устройствами.

› Корпус для полетов внутри помещений выполнен так, чтобы не перебить ценные вещи.

По выдаче от **netstat** мы увидим, есть ли открытые порты, и если да, то какие и к какому адресу они привязаны. Старый добрый **netstat** отлично подходит для отслеживания используемого приложением порта и отладки сетевых проблем. Давайте посмотрим, что тут можно найти.

Команда **netstat -p** показывает, что у нас есть открытый порт TCP на 5559 и три UDP-порта – 5554, 5555 и 5556, все принадлежащие процессу **program.elf**. С учетом функциональности AR.Drone, на это есть целый ряд возможных причин. Один порт нужен для управления самим устройством, еще два – для видеоканалов передней и нижней камер. Это охватывает UDP-порты, но чтобы более точно определить, для чего какой порт предназначается, следует смотреть более пристально.

Примечательно, что единственным механизмом безопасности, встроенным в устройство, является техника спаривания устройств, доступная в настройках приложения: *Free Flight*. Ее нужно включать для предотвращения вредных хулиганов по Wi-Fi, которые могут захватить контроль над устройством, когда оно находится в полете.

Высокий полет

AR.Drone великолепно оборудован – имеет встроенный Linux, много оперативной памяти и надежную механическую часть для обеспечения полета. То, что на самом деле открывает ряд возможностей для этой удивительно захватывающей игрушки, доступно на встроенной установке Linux BusyBox. С копией Linux, летающей под вашим контролем, вы получаете возможность делать много необычных вещей.

Выполняя простые сканирования портов и проверки открытых портов и процессов, можно передавать пакеты туда и обратно в устройство, что потенциально позволяет создавать потоковое видео и механизмы контроля, которые используются на выбранном языке или платформе.

Чего бы я ожидал от сообщества AR.Drone в будущем – это основное приложение с открытым кодом для управления устройством. Это открыло бы железо на использование любого варианта Linux. А пока что это одна из слабых сторон, которая не должна мешать развитию одной из самых увлекательных современных игрушек. 

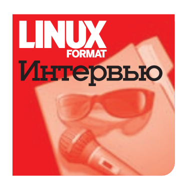


› AR.Drone с насаженным корпусом для запуска вне помещений.



Мэри Колвиг, маркетолог сообщества Mozilla, беседует с Джоно Бэконом об инновациях, культуре и многом другом.

Служение сообществу



Мэри Колвиг [Mary Colvig], живущая в Сан-Франциско, работает в Mozilla Project с июня 2006 года. Мэри была движущей силой роста и вдохновения сообщества Mozilla, и принимала активное участие в попытках сделать запуск *Firefox 3* мировым рекордом, внесенным в Книгу рекордов Гиннеса. В настоящее время она считает программу событий сообщества Mozilla частью своей роли в маркетинговой политике сообщества.

Мы уселись поговорить с Мэри о сообществе, о месте *Firefox* в мире *Internet Explorer* и *Chrome*, мобильных устройств, социальных медиа и т.д.

LXF: Как вы очутились в Mozilla?

MK: Все началось с забега! Я много лет помогала некоммерческой организации Leukemia & Lymphoma Society [Сообщество Борьбы с Лейкемией и Лимфомой] на добровольной основе, занималась поиском фондов и работала в их команде. После многолетних тренировок и участия в соревнованиях по триатлону от Leukemia & Lymphoma, я присоединилась к тренировкам их спортивной команды Ironman для участия в Ironman Canada 2005.

Моя одержимость историей *Firefox* и Mozilla росла, и тут я начала бегать с Рафаэлем Эброном [Rafael Ebron], который тоже был членом команды Ironteam от Leukemia & Lymphoma Society. Я узнала, что он работает в Mozilla, и для меня это был просто шок. Я даже не думала никогда, что там работают реальные люди, и точно так же не представляла себе, что есть такое «там». Мы с Рафаэлем очень скоро стали бегать вместе, и он рассказывал мне про Mozilla и *Firefox*.

LXF: Так значит, ваш забег в итоге привел вас на работу в Mozilla?

MK: Ну, в начале 2005 года Рафаэль рассказал мне, что Mozilla собирается создавать новую PR-команду. Это была чудесная возможность поработать не просто с моим любимым продуктом, но и с великой миссией с упором на некоммерческую деятельность. Это объединяло все мои пристрастия в одно, и кончилось тем, что я возглавила внешнюю PR-команду Mozilla.

В 2006 году я вошла в оплачиваемый штат, округлив нашу маркетинговую команду до двух человек. Это было чудесно – найти свой собственный способ содействия через пропаганду и маркетинг.

LXF: Чем вы там занимаетесь?

MK: Сейчас я директор по вовлечению добровольных помощников в Mozilla. Цель нашей команды – дальнейшее укрепление самого сообщества и поддержка членов сообщества Mozilla. Это люди, которые уделяют свое время и знания и умения проекту, а также продвигают и популяризируют Mozilla и наши принципы. За эти годы мы наблюдали невероятный и разносторонний рост сообщества, но мы можем сделать возможность сотрудничества с Mozilla и соответствующую поддержку еще проще.

В этом году мы занялись выяснением, как люди узнают о возможности сотрудничать с Mozilla и как сделать процесс этого сотрудничества более гладким и ровным, поделиться инструментами и ресурсами, которые могут облегчить жизнь нашим помощникам, добавить наставничества и способов измерить объем оказанного содействия и здоровье сообщества в целом.

LXF: Есть ли типичный день в вашем рабочем ритме в такой компании, как Mozilla?

MK: Нет, типичных дней определенно нет. Мне приходится переключать скорость и менять обличье много раз в день. День обычно начинается рано, а заканчивается поздно, поскольку я работаю с членами сообщества по всему миру.

LXF: Какого рода содействие оказывает сообщество Mozilla?

MK: Люди у нас занимаются практически всем – кодом, дизайном, поддержкой пользователей, тестированием, даже берутся за метрику и юридические вопросы.

Мы работаем над расширением способов участия всех пользователей в улучшении *Firefox*. В серии *Firefox 4* beta мы развернули проект Feedback Add-on от Mozilla Labs Test Pilot (http://www.mozilla.com/en-US/legal/privacy/firefox.html), позволяющий принять участие в анонимном исследовании, которое нацелено на усовершенствование работы *Firefox*. К Test Pilot присоединилось более миллиона человек.

Мы также запустили программу поддержки сообщества под названием Army of Awesome (http://support.mozilla.com/en-US/army-of-awesome), которая позволяет фанатам *Firefox* общаться друг с другом и отвечать на вопросы по поддержке в Twitter. В течение первых недель сотни добровольцев уже начали отвечать на тысячи вопросов пользователей в Twitter.

LXF: Как, по-вашему, будет изменяться и развиваться сообщество Mozilla в последующие годы?

MK: Сообщество значительно выросло в масштабе и стало более разнообразным в плане географического распространения и компетентности. В плане охвата, в последнее время новые сообщества появились в таких местах, как Юго-Восточная Азия и Африка. По статистическим данным, у нас появилось более 60 членов сообщества в Кении, и эта команда проводит ежемесячный Mozilla Day. Они создали сайт своего сообщества (http://mozilla-kenya.org), чтобы делиться своими достижениями и поощрять других присоединиться к их сообществу.

О СОДЕЙСТВИИ

«Мы расширяем способы участия всех пользователей в улучшении Firefox.»

В плане компетентности, наша база помощников увеличилась, включив не только обычных добровольцев по работе с кодом, локализацией и QA, но и людей творческих и даже экспертов по метрике. Теперь у нас есть десятки тысяч дизайнов Persona или легковесных скинов для *Firefox*, которые отражают практически любое хобби или увлечение. Также мы недавно завершили соревнования по визуализации открытых данных, чтобы ответить на вопрос, как люди используют *Firefox* (http://blog.revolutionanalytics.com/2011/01/winners-of-mozilla-open-data-competition-announced.html).

LXF: Конкуренция среди web-браузеров все растет и растет — вы полагаете, что Mozilla ей соответствует?

MK: Безусловно! Наша миссия заключается в стимуляции среды, в которой пользователи пожинают плоды последовательных инноваций. Мы продолжаем внедрять инновации в настольный браузер и почту, набирая силу в мобильных устройствах, приложениях и в сфере обмена данными и сервисами, такими, как Sync.

Кроме того, что мы освоили солидную долю рынка, наше сообщество выросло, становясь все более энергичным и обширным. Объединяя наше сообщество и открытые web-технологии, мы можем поощрять все больше людей к деятельности в сети. Здесь прекрасным примером служит Universal **Subtitles, Mozilla Drumbeat Project, который дает людям возмож**ность сотрудничать и создавать субтитры на множестве языков к самому разнообразному контенту. На прошлой неделе мы наблюдали, как оно сработало при переводе Обращения к Конгрессу Президента США [State of the Union] (http://brettgaylor.tumblr.com/post/3026870192/yes-we-did).

LXF: Каким бы вам хотелось увидеть рост сообщества Mozilla?

MK: Работая в общественной организации, я бы хотела, чтобы у всех, кто с любовью относится к Mozilla и/или *Firefox*, была возможность оказать им свою помощь и почувствовать себя своего рода акционером. Это означает исследование новых возможностей, доступных более чем для 400 миллионов наших пользователей. LXF

Работа вашей мечты

Джоно Бэкон, менеджер сообщества Ubuntu, делится экспертным мнением о том, как начать продвижение по карьерной лестнице свободного ПО.





Приди мне в голову держать пари, я бы поспорил, что большинство читателей *Linux Format* с радостью работали бы в индустрии открытого кода. Многие из вас обожают Linux и свободное ПО, а что может быть лучше, чем сделать на своем хобби еще и карьеру? Возможно ли получать зарплату за любимое занятие?

В этой статье я хочу поделиться советами и рекомендациями о том, как получить работу вашей мечты, связанную со свободным ПО. Я рассмотрю доступные возможности и способы выделить свои таланты и свое резюме из массы. Я также опишу мой собственный опыт, приведя анекдоты, способные помочь вам приблизиться к получению оплачиваемой работы в области открытого кода.

Вот вам хорошая новость: мир свободного ПО бурлит возможностями для профессий всяческого рода. Не опасайтесь, что компании ищут только программистов на C: это вовсе не так. Программисты, художники, дизайнеры, писатели документации, тестеры, специалисты по развитию бизнеса и продажам, а также многое другое – все это варианты, доступные для вас на рынке труда. Проблема не в поиске подходящей работы, а в обретении всего того, что работодатель желает видеть у кандидата на вакансию.

Одной из причин, по которым я охотно взялся за написание этой статьи, было то, что мне повезло получить работу моей мечты, связанную с открытым ПО: я – менеджер сообщества Ubuntu. Мало того, у меня есть две другие работы-мечты: журналиста и профессионального консультанта по свободному ПО. Выходит, найти достойную работу можно; главное – приложить для этого усилия и упорно трудиться.

Поиск работы

Первым делом создайте себе твердое представление о предложениях рынка: каких они требуют обязанностей, интересны ли вам и совпадают ли с вашими навыками. Я бы советовал вам выбирать работу, расширяющую ваши горизонты, но работодателю нужно знать, что у вас есть солидный фундамент для данной роли. Если вы собираетесь, получив место, обрести все необходимые знания по ходу дела, вам придется-таки побарахтаться – если только вы не впитываете информацию как губка. Во избежание этого, всегда ищите предложения, подыгрывающие вашим исходным навыкам.

> «Работодателю нужно знать, что у вас есть солидный фундамент.»

Чтобы найти открытую вакансию, просмотрите сайты компаний, где вы жаждете работать. Пусть это поставщик дистрибутивов Linux; тогда проверьте страницы Карьера в Canonical, Red Hat, Novell, Mandriva и других. Отметьте интересующие вас позиции и прочтите соответствующие требования. Мой первый совет – читайте их внимательнейшим образом. Ничто не бесит более, чем отклик от кандидата, который явно не читал описание работы. Кроме того, честно оцените, насколько вы под требования подходите – если работодатель требует пяти лет опыта программирования на Python, а у вас есть только год, не заявляйтесь на вакансию, воображая, что сумеете заставить компанию передумать. Вы же не хотите попасть в черный список для будущих предложений от этого работодателя.

Сайты о трудоустройстве – это второй источник для поиска работы. Имейте в виду, однако, что многие наниматели обращаются к ним только тогда, когда публикация на сайте компании дала мало откликов от достойных кандидатов. В такой ситуации вы можете увидеть, что менеджер по найму персонала чуть сильнее озабочен заполнением позиции, но зато и откликов от кандидатов, скорее всего, гораздо больше. Поэтому я всегда рекомендую отслеживать публикации новых вакансий на сайтах компаний.

Опыт

На том или ином этапе своей карьеры многие из нас сталкиваются с одной и той же загвоздкой: любая работа, оказывается, требует большого опыта, а если такового у вас нет, как же его без работы приобрести?

К счастью, в нашей области для вас открыты величайшие возможности. По самой природе сообщества свободного кода, любого желающего с радостью примут в его разработку. Если вы новичок, вам всегда кто-нибудь поможет научиться основам. Если у вас нет должного опыта для работы вашей мечты, самое время его получить, а свободный код – прекрасное место для этого.

Расмотрим такой пример: вот вы хотите быть программистом. Есть тысячи проектов, вопиющих о помощи. Вливайтесь, и ваш кусок кода попадет в проект. Накопив опыт по проекту, разработайте новую функцию и затем ее внедрите. Все это будет засчитываться как опыт программирования и отлично впишется в ваше резюме.



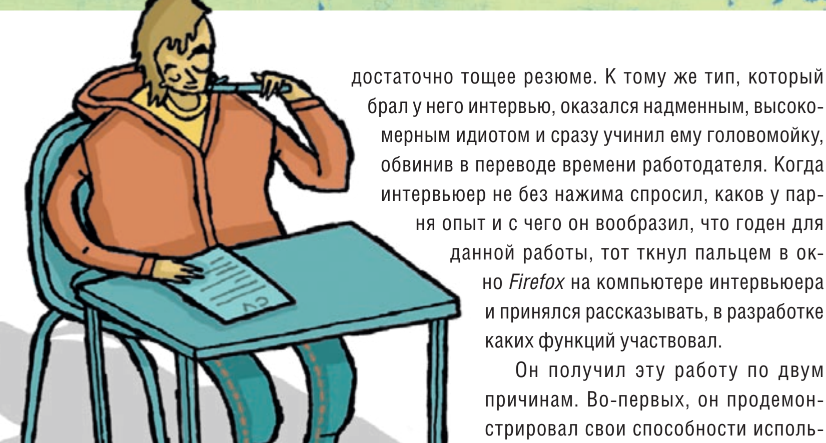
Это напоминает мне историю, которую я слышал пару лет назад, про умного парнишку, отчаянно пытавшегося найти работу. У него не было опыта коммерческого программирования, школу он бросил в 16 лет и имел

Нужно ли высшее образование?

В связи с работой в индустрии свободного кода, меня часто спрашивают о критериях по образованию. Нужен ли университетский диплом? Нужна ли степень магистра или сертифицированная квалификация инженера?

Критерии обычно варьируются в зависимости от вакансии, но в целом диплом требуют все реже и реже. Я не призываю студентов сбегать с обучения компьютерным наукам, но и без диплома не стоит чувствовать себя за бортом. Наличие большого опыта и знаний по конкретным технологиям, необходимым для открывшейся вакансии, обычно побивает университетское образование.

Единственным исключением здесь являются позиции администраторов и тренеров, где обязательны сертификаты в предметной области (например, RHCSA и RHCE для системных администраторов Red Hat). Может показаться заманчивым вложиться в курсы по сертификации, но они стоят весьма недешево, и стоит сначала проверить, сколько рабочих мест они вам откроют. Вы прекрасно продемонстрируете свой опыт, участвуя в проектах с открытым исходным кодом.



достаточно тощее резюме. К тому же тип, который брал у него интервью, оказался надменным, высокомерным идиотом и сразу учинил ему головомойку, обвинив в переводе времени работодателя. Когда интервьюер не без нажима спросил, каков у парня опыт и с чего он вообразил, что годен для данной работы, тот ткнул пальцем в окно *Firefox* на компьютере интервьюера и принялся рассказывать, в разработке каких функций участвовал.

Он получил эту работу по двум причинам. Во-первых, он продемонстрировал свои способности использования свободного кода и разработки сообщества как плацдарма для улучшения своих навыков и помощи проекту. Он создал реальный код, который был включен в продукт для миллионов пользователей, и работал с людьми и процессами ради достижения этой цели. Но это было лишь половиной того, что произвело впечатление на интервьюера: другая причина была в показе, что он сам нашел для себя дело и набрался в нем опыта. Парень выглядел уже не просто талантливым программистом, а «золотым билетом», который ищут все менеджеры по найму: сочетание личной и профессиональной мотивации.

Это одна из чудеснейших возможностей, предоставляемых свободным и открытым исходным кодом. У нас есть глобальная сеть проектов, которая не только обеспечивает потрясающее место для изучения новых технологий и сотрудничество с лидерами в этих областях, но также доставляет средства демонстрации вашей личной мотивации. У вас есть возможность обогатить свою жизнь, постоянно улучшая свободное ПО открытого кода для других пользователей.

Итак, вот лучший совет, который я могу дать вам в этой статье: идите и вносите свой вклад в проекты с открытым кодом. Включайтесь в разработку, участвуйте в общей дискуссии (это покажет, как вы взаимодействуете с другими людьми), профессионально и вежливо разрешайте конфликты и разногласия – и создайте постоянно растущий список проектов, которые можно привести как примеры вашего вклада. Возможность прийти на интервью и сказать, что миллионы людей используют ваш код, уже вносит вас в категорию заявителей, достойных серьезного рассмотрения.

Этому совету сопутствует предостережение: если вы намерены участвовать, не забывайте о значимости и устойчивом характере вашего вклада. Предоставление 100 примеров проектов, где вы внесли несколько мелких улучшений, а потом вам это надоело и вы перескочили на другой проект, не произведет желаемого эффекта. Вам нужно доказать, что вы внимательны к деталям и умеете справляться с реальными проблемами в долгосрочной перспективе, а вклад временного попутчика такого эффекта не даст. Будь я интервьюером, на меня бы произвели гораздо большее впечатление два-три примера качественных вкладов в проекты, предоставивших значительную ценность и сделанных с уважением к коллегам по сообществу. Это шлет гораздо более сильный и позитивный посыл.

Стоит также отметить еще один важный плюс длительного и результативного участия в проектах: на этом строится ваша репутация. Множество людей получают работу своей мечты, потому что имеют положительную репутацию благодаря большой работе с сообществом. Масса моих знакомых начинали с IRC-каналов и списков рассылки, способствуя проектам с открытым исходным кодом. После чего им часто предлагали работу благодаря сложившейся репутации в сообществе и их достижениям.

Показ профессионализма

В современном мире социальных сетей, где мы окружены ореолом информации, традиционный подход к разделению профессиональной и личной жизни ушел в небытие. В старые времена вы были в профессиональном режиме, когда одевали костюм, а когда костюм снимался, наступало время для пятерки уровней *Sonic The Hedgehog* и коробки вина. Работа и досуг четко различались. Сегодня все иначе. Большинство из нас используют Facebook, Twitter, Identica, Linked In и другие сайты, и во многих из них эти два режима переплетаются.

Facebook – отличный пример такого слияния. Я знаю много людей, которые появились на Facebook с четкой целью добавлять только друзей и семью. Но кабы все было так просто! А как же быть с коллегами? Если коллега добавил вас в друзья на Facebook, будет неловко, если вы проигнорируете его запрос, объявив, что коллеги – это не друзья. Таким образом, обычно довольно закрытая страница профиля в результате наполняется коллегами, размывая грань между ними.

Задать щекотливые вопросы

Многие люди ощущают дискомфорт при выяснении вопросов о зарплате на позиции и компенсационном пакете. Когда это лучше делать, и как это делать? Спросить ли об этом по электронной почте? Или во время интервью? Что делать, если зарплата слишком низка или вас не устраивает компенсационный пакет – например, нет медицинского страхования для вашей семьи?

Наилучший момент, насколько я могу судить, побывав в положении обеих сторон на интервью, возникает тогда, когда после прохождения всех интервью компания предлагает вам позицию. На этом этапе вы должны поставить вопрос вежливо, откровенно и ребром. Если компания еще не спрашивала вас о так называемых компонентах компенсационного пакета, спросите, какая зарплата предлагается для этой позиции.

Если предложенная зарплата мала – но учтите, что ради получения перспективной работы в стоящей компании можно и потерять в деньгах – не стесняйтесь просить больше. Выгода обсуждения зарплаты после получения места состоит в том, что ваши потенциальные наниматели, как вы уже поняли, хотят взять на работу именно вас и знают, что позиция может быть заполнена, хотя бы и с дополнительными затратами. Искать персонал никто не любит, и чтобы снова не затевать этот процесс ради кандидата подешевле, компания может проявить гибкость и пойти вам навстречу.



Я всегда был того мнения, что нужно быть самим собой, и на работе, и дома. Я всегда был рад поделиться моей личной жизнью профессионально, и наоборот. Мне до сих пор есть чему поучиться в этом мире, но я рад, что мои профессиональные коллеги и общество в целом знают, что я слушаю металл, люблю свою жену, и мне нравится играть концерты со своей группой. Отключить эти характеристики я не могу – они являются определяющими компонентами и делают меня тем, кто я есть. Хитрость заключается не в разделении людей из вашей жизни на различные группы, а в разумном и сознательном общении с группой в целом и в разных ситуациях.

Это тем более важно в мире открытого исходного кода. Сообщество свободного кодом именно и есть сообщество. Это набор профессиональных отношений, но и множество связей, отлитых и развитых в общественных мероприятиях и других случайных средах. Наша общительная природа – вот что делает наше сообщество таким замечательным; не старайтесь скрывать это в профессиональной среде.

Резюме: дверь в мир

Став менеджером и начав нанимать людей, я ощущаю преимущество знания, что такое быть по другую сторону стола во время интервью. Я почти хотел бы, чтобы каждый имел возможность узнать, что значит нанимать людей, потому что это помогает понять и исправить свои недостатки при приеме на работу. Такую возможность имеют не все, и я поделюсь наиболее важными вещами, на которых надо сосредоточиться при поиске работы вашей мечты.

Первым пунктом, безусловно, является ваше резюме. Поскольку отрасль открытого кода популярна и найдется множество людей, желающих найти их работу мечты, в ответ на опубликованные вакансии мы часто получаем большое количество откликов. На каждую позицию, на которую мне приходилось нанимать людей, выстраивалась очередь претендентов (Canonical – фирма, где хотят работать многие) – и резюме становится первой точкой контакта с работодателем. Я использую его, чтобы решить, приглашать ли человека на интервью или нет.

Резюме приходят в различных формах. Одни умещаются на листе бумаги, другие представляют собой практически роман. Одни скучны и суховаты, другие – ярки и красочны. Хитрость тут в получении правильного баланса.

Кричащее и яркое, но малосодержательное резюме не цепляет. Равно как и нудное резюме, хотя бы и с тоннами содержания. Некоторые резюме пестрят орфографическими ошибками и прочими ошибками новичков. Я рекомендую придерживаться следующих принципов:

» Делайте резюме кратким и точным. Охватите все ключевые факты, которые вы хотите донести до работодателя, но не перегружайте чрезмерным количеством деталей. Представьте лучшие свои характеристики, честно и четко; избегайте жаргона.

» Убедитесь, что выделены важные моменты. Лично я читаю все получаемые резюме от начала до конца, но знаю, что многие менеджеры просто пробегают по ним глазами. Чтобы избежать этого, убедитесь, что ключевые элементы информации – ваши навыки, достижения в области образования и трудовой стаж – выделены жирным шрифтом, чтобы их отметили.

» Опишите основные элементы. Резюме – это документ, цель которого – представить вас и ваши навыки, которые вы можете предложить потенциальному работодателю. Убедитесь, что вы даете им всю необходимую информацию. Вы найдете онлайн много материалов по созданию хорошего резюме.

» Удалите все следы эгоизма. Мне приходилось читать резюме, сплошь заполненные самовосхвалением. Это не дает позитивный сигнал. Конечно, ваше резюме призвано продать вас, но пусть ваши навыки и опыт говорят сами за себя – вместо того, чтобы превращать их описание в длинное банальное коммерческое объявление.

» Не лгите. Это кажется очевидным, но стоит повторить: не лгите. Даже если вам это удастся на стадии отбора по резюме, интервью, как правило, раскрывает любую неправду. Поимка вас на вранье угрожает вашим будущим шансам.

«Кричащее и яркое, но малосодержательное резюме не цепляет.»

Созданное вами отличное резюме выведет вас на интервью!

Совет на этой стадии процесса найма прост: будьте скромны, честны, и отвечайте на вопросы как можно лучше. А когда интервьюер спросит, есть ли у вас вопросы к ним, используйте это как возможность больше узнать о позиции. Всегда держите в уме по крайней мере два или три готовых вопроса.

Найти работу своей мечты – это обширная тема, и на полное ее раскрытие здесь нет места, но я надеюсь, что мои советы помогут вам продвинуться в этом направлении. Удачи! LXF

Наполните свое резюме

Есть много прекрасных вещей, которые вы можете сделать, чтобы укрепить свою репутацию и опыт в вашей карьере. Вот лишь несколько идей:

» Опубликуйтесь – не желая увеличить и без того бездонное море писем по электронной почте в башни Linux Format; публикация в журнале, на сайте или в других местах – это отличный способ добавить шарма вашему резюме.

» Напишите книгу. К этой рекомендации стоит отнестись с осторожностью. Писать книгу – неблагодарный труд. Он включает многочасовой труд, стопки исследований и бесконечные редактирования – и все это одновременно с основной работой и поддержанием здоровых отношений с друзьями и семьей. Но, несмотря на все мучения, это делает сильную заявку, что вы обладаете определенными знаниями в той области, по которой написана ваша книга.

» Организуйте мероприятие: организация мероприятия, конференции и тому подобного сборища – отличный способ продемонстрировать управление взаимосвязанными частями. Многие компании часто и сами хотят организовывать события, так что наличие такого опыта может оказаться хорошим бонусом.

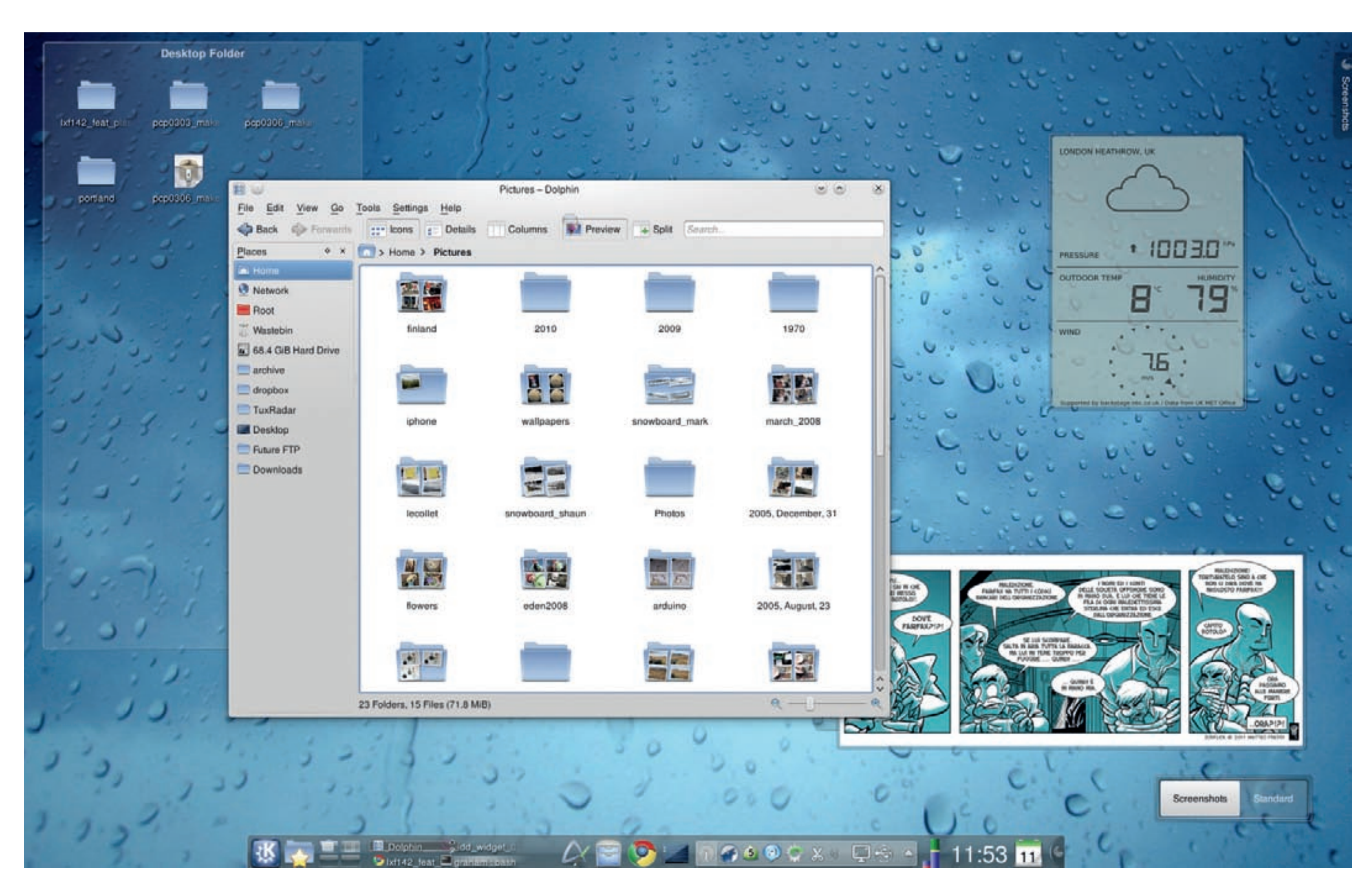
» Зарегистрируйтесь в группах и сообществах: участие в местных группах и сообществах, типа Групп Пользователей Linux – еще один показатель природной жажды знаний и двигатель дальнейшего развития карьеры.

» Будьтесь общительны – больше используйте Twitter, Facebook, LinkedIn и другие социальные сети, чтобы начать строить репутацию для выполняемой вами работы.

Где бы вы ни унюхали возможность любого рода для развития своего портфолио или налаживания контактов, немедля ловите ее!







Осваиваем Рабочий стол Plasma в KDE

Амбициозный Рабочий стол Plasma в KDE готов к использованию; **Грэм Моррисон** поможет раскрыть его достоинства.

При первом объявлении о KDE 4, главным его свойством был переход от привычного поведения — когда значки загромождают рабочий стол, а приложения запускаются из одного меню — к тому, что разработчики KDE назвали Рабочий стол Plasma [The Plasma Desktop]. Эту концепцию анонсировал Аарон Сейго [Aaron Seigo] в 2005 году. Планировалось, что бесплотный покров Plasma заменит многие элементы, видимые на экране, и что функциональные фрагменты смогут перемещаться, изменяться и удаляться по прихоти пользователя. Например, значки на рабочем столе появлялись только через вызов специального виджета, а панель запуска и меню запуска заменили на смешанный виджет. На достижение этого ушло пять лет.

Главной проблемой этих новшеств до недавнего времени было то, что ни одно из них не упрощало работы, то есть не создавало особых причин им пользоваться. По сути, Plasma часто усложняла простые задачи,лишая их стабильности, так как во многих местах технология давала сбои. Сейчас, когда уже вышли релизы KDE 4.5 и 4.6, команда KDE, наконец, свела концы с концами. Они не только перестали совать слово Plasma в каждую задачу, но и соорудили панель запуска и движок для виджетов, а целое множество тем и действий [activities] рабочего стола стали намного доступнее и стабильнее, превратив Plasma в реальное преимущество пользователей KDE. Поскольку набор функций приблизился к некоторым смелым идеям, изначально объявленным для KDE 4.0, пришло время удалить виджеты *SuperKaramba*, забыть про скрипты DCOP и присоединиться к революции Рабочего стола Plasma.

> «Plasma усложняла простые задачи, лишая их стабильности.»

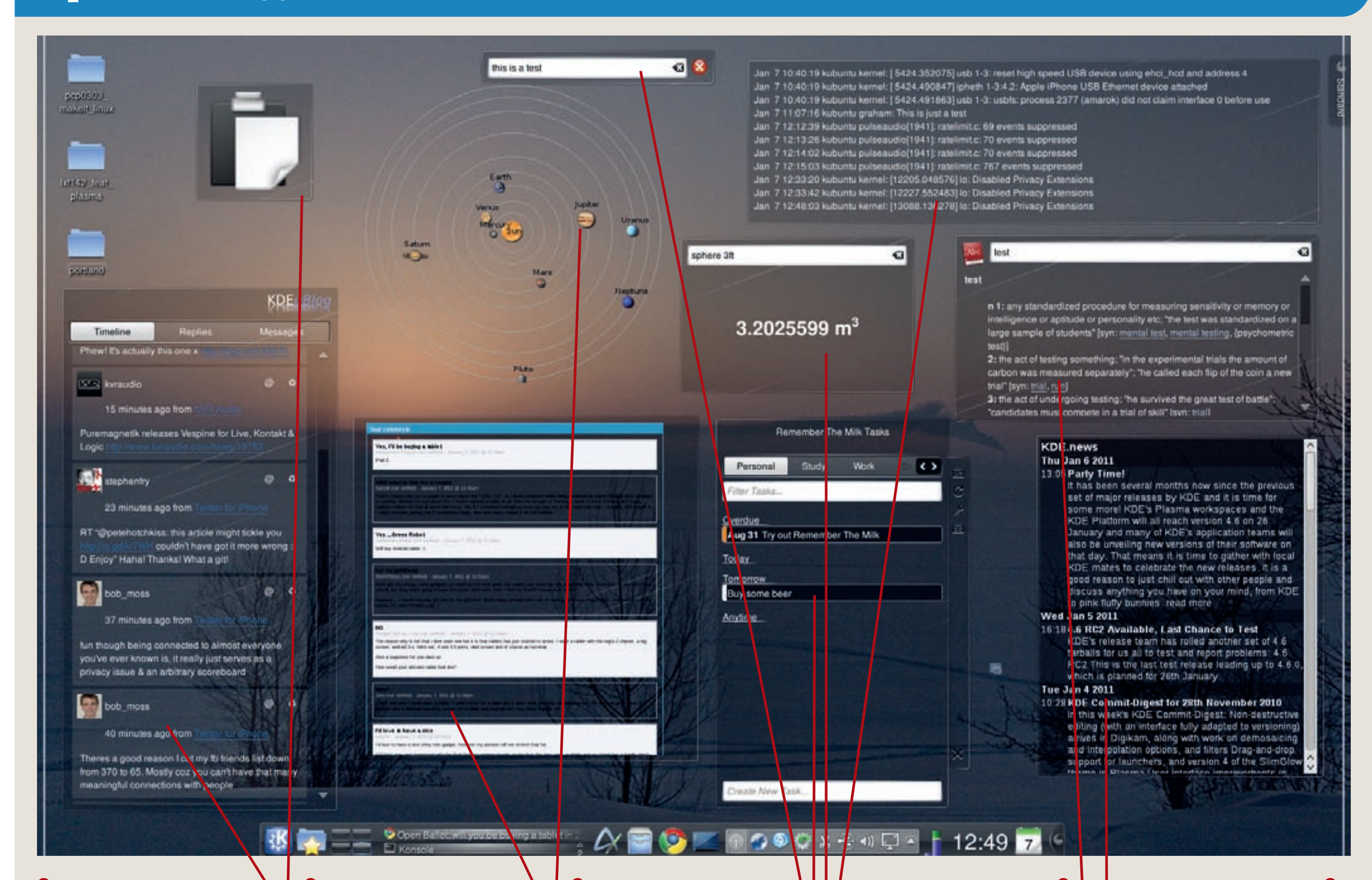
Первое, что следует узнать про Рабочий стол Plasma – когда и как делать изменения в этом рабочем столе. По умолчанию сразу начать изменять его нельзя. Сперва его нужно разблокировать, что достигается правым щелчком мышью либо по небольшому значку Plasma справа на панели задач, либо по панели Plasma, прикрепленной к правому краю рабочего стола.

Результаты будут одинаковы: дозволение добавлять виджеты, действия и новые панели, но только при разблокированном рабочем столе. Именно поэтому, например, нельзя просто перемещать значки по рабочему столу: они будут упорно цепляться за свое место, пока их не разблокировать. Но капризы функции блокировки также отвечают за поведение фона рабочего стола. Например, можно включить старый просмотр файлов и папок, отобразить простой рисунок или даже включить вывод последних новостей.

Эти возможности спрятаны в выпадающем меню панели настройки Desktop Settings, но способ попасть к этой панели непредсказуем. Нужно щелкнуть правой кнопкой мыши по рабочему столу и выбрать последнюю опцию меню. Название опции зависит от текущего действия на рабочем столе, а изменению это действие поддается, только если вы ранее разблокировали рабочий стол. Разблокировка при открытом окне не работает. Оба этих ляпа, похоже, последние остатки колоссального списка проблем в KDE 4.0, и хотя их присутствие в KDE 4.5 ничуть не более допустимо, с прочими функциями Plasma у вас подобных проблем уже не будет.

Установка Folder View в панели Desktop Settings решает другую проблему – источник жалоб многих пользователей после появления KDE 4.0, а именно: где использовать фон рабочего стола как импровизированный файловый менеджер. Дело в том, что по умолчанию рабочий стол KDE только создавал ссылки на файлы и папки, якобы помещаемые на него, а самих файлов на нем не было. И если вы хотели скинуть файлы в папку на рабочем столе, а затем, закончив работать с ними, поместить их в архив, вас ждала неудача. Включение режима Folder View возвращает рабочий стол к своей первоначальной функции, когда отображалось настоящее содержимое папки Desktop вашего домашнего каталога, с возможностью перемещать, переименовывать и удалять файлы. Такой переход стал возможным благодаря виджету »

Лучшие виджеты KDE



Microblogging
Это приложение для неувядающих Twitter и Identica. Оно ломится от функций и занимает примерно столько места, что и 140 символов.

Web slice
При вводе URL отображает ту часть сайта, который вы хотите отобразить и обновить, что стоит после элемента тэга **<div>**.

KRunner
На самом деле это не виджет, но при нажатии Alt+F2 вы сможете быстро найти подходящий виджет и запустить нужную программу.

File watcher
Его можно использовать для просмотра изменений в каком-либо файле, что особенно полезно при отслеживании локальных и удаленных файлов журнала.

News feed
Простой автоматически обновляющийся RSS-агрегатор. Можно настроить, чтобы в нем отображался только заголовок, а можно, чтобы показывалось содержимое новости.

PasteBin
Делитесь своими текстами и изображениями с помощью учетной записи PasteBin, просто перемещая файлы на этот удобный значок.

Solar system
Ну да, это просто украшательство, но оно лучше, чем виджеты выпученных глаз или лунных фаз.

Remember The Milk
Это замечательный список дел, основанный на облачных данных. Его можно синхронизировать, чтобы следить за самыми срочными делами.

Qalculate
Прекрасная математическая программа, способная решать уравнения и вычислять такие значения, как, например, радиус сферы.

Dictionary
Держать словарь на своем рабочем столе – полезный способ развить свой словарный запас и избежать ошибок.

Plasma, который активируется и устанавливается на работу в полноэкранном режиме. И сейчас мы перейдем в мир виджетов.

О виджеты, зачем вы?

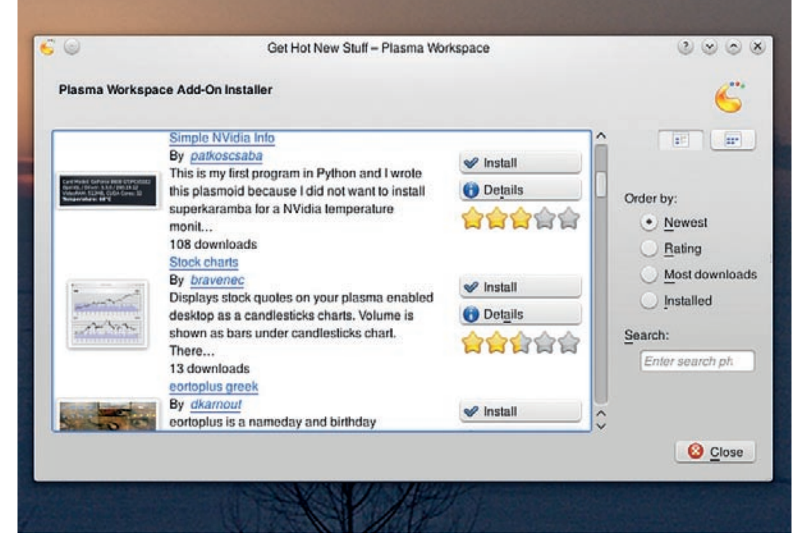
Говоря о Plasma, средний пользователь KDE обычно подразумевает виджеты. Они прославлялись как главный повод обновить KDE 3.5, заменив другие, несравнимые технологии виджетов (включая *SuperKaramba*), на единую, официальную систему. Изначально обещалась широкая совместимость с виджетами от Google и Apple, но это, а также некоторые части дизайна виджетов, все еще необходимо внедрять в Plasma – набор ныне доступных виджетов – чтобы наверстать дефицит.

Существует два типа виджетов Plasma: прикрепляемые к рабочему столу и прикрепляющие себя к панели. Разработчики KDE предприняли умный подход к этой дуальности. Если виджет совместим с обоими типами, его можно перетаскивать с рабочего стола на Panel, и он изменит свой вид. Перетащите виджет Calendar на фон рабочего стола, и вы получите обзор всех дней текущего месяца. Перетащите его на Panel, и увидите одну дату в значке перекидного календаря. Именно по этой причине есть два способа добавить виджет на ваш рабочий стол: первый – нажатием кнопки на правой стороне экрана, а второй – нажатием на виджете Plasma, прикрепленном к правой стороне панели задач. Оба откроют библиотеку виджетов, которая выглядит как горизонтальный список виджетов с прокруткой внизу экрана. Список можно фильтровать по категории или использовать поле поиска для нахождения виджета, содержащего определенную строку.

> «Можно получать доступ к файлам безо всяких меню.»

Если список виджетов по умолчанию недостаточен для ваших нужд, кнопка Get New Widgets теоретически может вам удружить. Но только в теории: пока нам не удалось заставить эту функцию работать. Кнопкой Local File можно скачать виджеты из раздела Plasmoids сайта www.kde-look.org, однако успех операции зависит от того, способен ли ваш дистрибутив удовлетворить специфические зависимости, требуемые виджетом. Хотя изначально разработчики KDE обещали поддержку виджетов Dashboard от Apple и к середине 2009 года была достигнута базовая совместимость, дальше дело не продвинулось, и сейчас эта функция зачахла, то есть в будущем для виджетов Dashboard места очень мало. Тем не менее, многие разработчики KDE все еще работают над виджетами, а нажав на кнопку Download Plasma Widgets, можно просмотреть онлайн-библиотеку виджетов, где перечислены те, что есть в репозитории. Любой заинтересовавший вас виджет устанавливается одним щелчком мыши.

› Многие программы KDE позволяют скачать новое содержимое из Интернета, и виджеты — не исключение.



Действия

Бывает, что выполнение одной задачи требует одного набора виджетов, а для другой задачи и набор нужен другой; но процедура добавления/удаления виджетов скучна и трудоемка. Функция Действие [Activity] в KDE решает эту проблему. Она позволяет создать глобальную настройку всего рабочего стола – вместе с виджетами, системными параметрами и фонами на ваш выбор – и присваивает эту настройку тому, что в KDE называется Действием. Можно создать Действие для редактирования, включив виджеты Kate и Dictionary, а также несколько приложений быстрого запуска для наиболее часто используемых вами файлов; другое же Действие приспособить для работы с Интернетом, и при необходимости переключаться между ними. Действия – нечто вроде расширения виртуального рабочего стола, причем каждый рабочий стол подстроен под определенную задачу; и создание действий и переключение с одного на другое в KDE реализованы максимально простым способом.

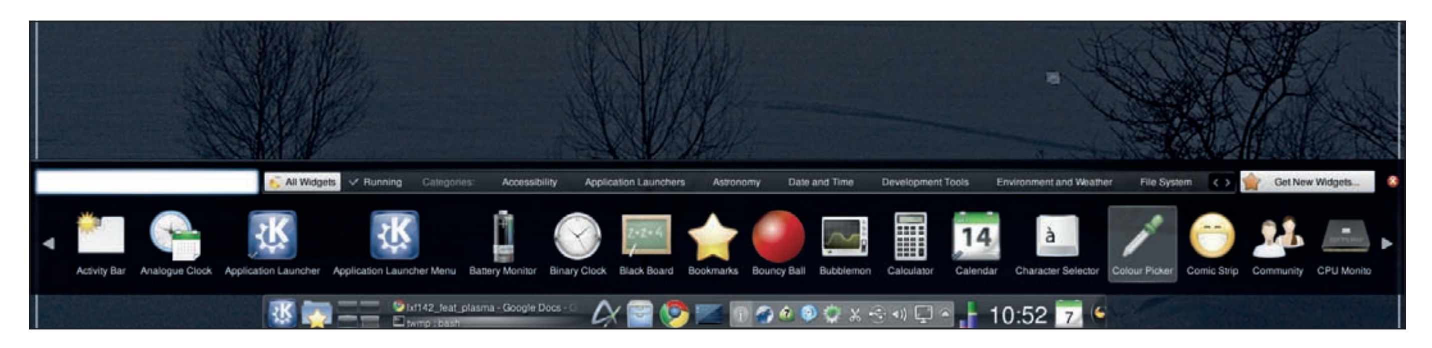
Палитра Activities внешне очень похожа на вид, используемый виджетами рабочего стола; она открывается по нажатию на Activities из значка Plasma в правой части экрана. Эта панель в KDE названа Plasma Activity Explorer, и чтобы создать в ней новую зону, нужно нажать на значок New Activity.

Вы увидите два значка-эмблемы [identicon] в списке на панели, по штуке для каждого Действия, и для переключения между ними достаточно щелчка. При смене Действия вы увидите, что фон рабочего стола, а также размещенные на нем виджеты, изменились согласно настройкам выбранного Действия. Чтобы ускорить переключения, Действия размещаются в памяти, но от этого можно избавиться, нажав на маленькую кнопку Stop слева вверху от каждой эмблемы. Это добавит в Действие значок Pause, отмечающий текущее состояние.

Одна из первых вещей, которые следует сделать после создания пары Действий – их переименование, чтобы избежать стоящего по умолчанию состояния Unnamed. В KDE такое проделать непросто: эта возможность спрятана в диалоге Plasma Workspace, на который мы уже жаловались. Он запускается при выборе последнего пункта меню щелчка правой кнопки мыши по фону рабочего стола. Мы не можем назвать вам этот пункт: он зависит от текущего режима фона рабочего стола. Также позаботьтесь о разблокировании возможности изменения Рабочего стола Plasma: если этого не сделать, окно Plasma Workspace только покажет имя Действия, а изменить его не позволит.

Переключение Действий

Так как Действия подобны виртуальным рабочим столам, переключение между ними лучше бы максимально упростить – например, с помощью комбинации клавиш; и KDE это позволяет. Щелкните по пункту меню Shortcut Settings, который появляется при нажатии на значок Plasma; откроется панель Workspace Configuration. По умолчанию можно вызывать появление Activity Explorer нажатием клавиши Meta (как правило, это клавиша с логотипом Windows) вместе с клавишей Q, но это сочетание можно изменить в окне горячих клавиш. Аналогично, если вы предпочитаете мышь клавиатуре, откройте окно Desktop Settings и перейдите в нем на вкладку Mouse Actions. Здесь присваивается переключение Activities на кнопку мыши или на действие мыши, но имеется одна неприятная загвоздка. Когда вы переключите Действие мышью, вы не сможете тем же способом перейти в другое Действие, не на-



› Перетаскивайте виджеты на рабочий стол или на панель. Многие будут автоматически менять свой размер.

страивая панель заново, если только в новом Действии переключение не выполняется тем же самым действием мыши. Более того, вам придется проделать ту же процедуру для всех Действий, между которыми вы желаете переключаться.

Такое поведение – следствие полной изоляции настроек одного Действия от других, и новое Действие, на которое вы переключитесь, не содержит соответствующей опции для мыши. Проблема станет более явной, если вы поместите виджет Activities на рабочий стол. Этот виджет позволит быстро переключаться между Действиями – либо по щелчку на названии, либо с помощью стрелок вправо и влево. Единственная проблема состоит в том, что если в Действии, на которое вы переключаетесь, этот виджет не установлен, из нового рабочего стола переключиться уже не получится. А если виджет установлен, но находится в другом месте, это может смутить еще больше: кажется, что он переместился из одного Действия в другое. Все это решается правильной настройкой, но было бы намного легче, если бы KDE сам заботился о таких вещах.

Панель

Последний основной компонент рабочего стола Plasma – панель Panel; она обычно служит для запуска приложений и переключения между запущенными программами. Мы сказали «обычно», поскольку она не обязана использоваться именно для этих целей; просто в установке KDE по умолчанию Panel прикреплена к низу экрана, и в ней содержатся меню Launch, Task Manager, раздел Quick Launch и часы. Можно спокойно удалить любой из этих пунктов, создать сколько угодно таких Panels и разместить их в любом месте рабочего стола. А все потому, что Panel – попросту виджет Plasma, только очень мощный.

Первым делом вы, скорее всего, захотите изменить размер панели, что достигается щелчком по значку Plasma в правой ее части. Тут откроется вид Resize, чуть похожий на выравнивание поля в текстовом процессоре. Стрелками налево и направо можно установить положение краев; перетаскивание другой стрелки отцентрирует Panel, а перетащив кнопку Height, вы установите размер по вертикали. По перетаскиванию кнопки Screen Edge панель переместится к другому краю экрана, и в этом же режиме можно перемещать любые компоненты, присутствующие на Panel. Например, Panel можно поместить на левый край экрана или на самый верх. Через меню правой кнопки мыши рабочего стола можно добавить столько Panel, сколько захотите, и незачем иметь одну монолитную Panel. Создайте несколько – каждую для своей задачи: например для меню Launch, Task Manager, часов или календаря.

Или вообще откажитесь от Panel. Благодаря *KRunner* можно получать доступ к файлам и приложениям в запуске безо всяких меню. Просто нажмите клавиши левый Alt и F2, и поверх экрана появится небольшое поле ввода *KRunner*. Начните набирать имя приложения или его функцию, и оно появится под полем ввода. Выберите нужную программу курсором и нажмите Enter. Часто это гораздо быстрее обращения к меню Launch, особенно если ваши руки уже на клавиатуре. *KRunner* также можно использовать для поиска и перетаскивания виджетов на рабочий стол. *KRunner* работает на базе набора модулей расширения, и их можно включать и выключать, нажав на маленький значок с гаечным ключом слева от поля ввода. Например, модуль Calculator позволит вводить числовые выражения прямо в поле ввода, и результат появится по нажатию на =. Модуль Power Control выключает или перезагружает компьютер по вводу команд shutdown или restart, а Web shortcuts позволит делать сокращения при поиске в сети и видеть результаты в браузере. Например, **wp:germany** отыщет в Википедии слово Germany; существуют и другие сокращения. Можно даже вводить команды *Bash*, останавливать процессы и открывать недавние документы. Как для всего Рабочего стола Plasma, пределы устанавливает только ваше воображение. LXF

Plasma на нетбуках



› В Рабочем столе Plasma есть специальный режим, «заточенный» на разрешение экрана нетбука.

Существует множество проектов, пытающихся создать идеальный интерфейс пользователя для малого размера экрана и мощности CPU среднего нетбука. Многие из них собраны с применением *GTK* и либо *Clutter*, либо *Compiz*, что осложняет для приложений KDE интеграцию с системой. Команда KDE разрешила эту проблему, создав собственный интерфейс пользователя из Рабочего стола Plasma, что дает поклонникам KDE возможность запускать свои стандартные приложения рабочего стола из родного окружения KDE.

Чтобы переключить рабочий стол KDE в интерфейс, дружелюбный к нетбукам, запустите приложение KDE Settings и щелкните по значку Workspace. В появляющемся окне можно менять глобальные значения для Рабочего стола Plasma, а в выпадающем меню Workspace Type – переключиться в режим рабочего стола нетбука. Через пару мгновений вы заметите, что фон рабочего стола сменился на набор значков для доступа к меню частых задач, и при запуске приложений они выполняются в полноэкранном режиме и без границ окна. Панель задач Panel появляется вверху экрана при наведении мышью и, как и в Ubuntu Netbook Edition от Canonical, отсюда можно переключаться между задачами, запускать новые приложения и вернуться к экрану запуска. Для пользователей KDE это лучший способ получить родную поддержку любимого окружения рабочего стола.

Что за штука... Gnome Shell?

Эндрю Грегори раздобыл снасти, запустил бородку, нацепил колпак и забросил удочку в будущее рабочего стола Gnome.

Разве мало оболочек для Linux: *Zshell*, *Xterm*, *Bourne Again Shell* и куча других! Я бы лучше что полезное почитал!
Да вас название сбивает с толку: *Gnome Shell* – совсем не оболочка, по крайней мере не та оболочка, о которой вы подумали. Это графическая система, которая лежит в основе Gnome 3.

А разве у Gnome есть нумерация версий? Это ведь у KDE постоянно новые функции и эффекты на уме — за Gnome такого не водится.
Именно поэтому перемены в Gnome так важны, а *Gnome Shell* заслуживает внимания. В общем-то цели *Gnome Shell* предельно прозаичны: разработчики стремятся помочь вам, убирая всё лишнее с рабочего стола, чтобы ничто не отвлекало от создания шедевров.

> «Всё, что не нужно для работы, собрали на отдельный экран.»

И что же уберут с моего рабочего стола?
Первое, что вы заметите – здесь нет ни апплетов, ни виджетов, словом, никаких привычных средств запуска приложений. Это хорошо хотя бы тем, что перед глазами нет значка *Firefox*, который так и зовёт бросить работу и отправиться в сказочное королевство Интернет. Обеспечивается максимум экранного пространства для рабочих приложений. Нет ни списка окон, ни перемещаемых панелей инструментов.

Кощунство! То, что вы говорите — это же безумие.
Нет, это трезвый расчёт. Годами Gnome практиковал такой подход, скрывая параметры настройки от пользователя в обмен на выбор разумных умолчаний. Менять настройки предполагалось только при необходимости, а не на каждом шагу. Та же философия сохраняется и в *Gnome Shell*. Если вы работаете над документом *OpenOffice.org*, *Gnome Shell* скрывает остальные работающие программы. Они по-прежнему легко доступны, но не лезут вам на глаза, требуя внимания.

Если мне понадобится минимализм, я возьму *Fluxbox*. Может быть, разработчики добавили что-то новое?
Конечно! Вообразите свой нынешний рабочий стол. Основное пространство занимает фон, который по мере открытия файлов постепенно заполняется окнами приложений. Остальная часть экрана (5–10 %, в зависимости от оконного менеджера) занята значками, меню и апплетами, позволяющими переключаться между задачами. Разработчики *Gnome Shell* собрали всё, что не нужно для работы в данный момент, на отдельный экран под названием Activity Switcher [Переключатель действий]. Доступ к нему открывается с помощью меню Activities, расположенного в верхнем левом углу зарана, или по нажатию клавиши Windows.

Ну вот, здесь уже есть смысл. Выходит, содержимое меню Applications, Places и System скрыто в Activity Switcher?
Не скрыто, наоборот – отображается во всех цветах радуги. Чтобы обеспечить ощущение преемственности между двумя вариантами вида, верхняя строка меню остаётся неизменной; примерно 80 % экрана внизу занято отображением открытых виртуальных рабочих столов. Слева находится колонка, где, сверху вниз, располагаются: поле поиска и меню Applications (в виде значков), меню Places и список документов, открытых последними. Любой элемент можно открыть двойным щелчком или перетащить на один из виртуальных рабочих столов. Чтобы вернуться в режим Activities, щёлкните на любом виртуальном рабочем столе, выберите пункт Activities в верхней строке меню либо нажмите клавишу Windows.

Знаете, что-то я затрудняюсь представить всё это…
Компоновку можно назвать логическим продолжением *Unity*: максимум полезного пространства, отображение меню в виде значков. Ну и замечательное оформление, конечно.

Нет, всё равно не могу.
Ну что ж, большинство дистрибутивов на основе Gnome уже содержит такие пакеты. Поищите *Gnome-shell* в своём диспетчере пакетов, введите команду **gnome-shell —replace** в терминале, и увидите всё воочию.

Упс! Не сработало, а машина зависла. Я начинаю ненавидеть этот *Gnome Shell*!
Да, у нас такое тоже пару раз случалось. Дело в том, что, по словам разработчиков Gnome, «…нет смысла ориентироваться на устаревшую аппаратуру. *Gnome Shell* поддерживает сравнительно примитивные возможности 3D-отображения, которые заложены практически во все компьютерные устройства, выпущенные за последние пять лет».

Эти сравнительно примитивные возможности включают аппаратное ускорение, и если ваш компьютер «не тянет» *Compiz*, он скорее всего не справится и с *Gnome Shell*. Но разработчики осознают, что такой поворот оставит сиротками изрядную долю потенциальных пользователей, и стараются устранить максимальное количество ошибок (не будем забывать, что разработка *Gnome Shell* ещё не завершена) и реализовать безотказный двумерный рабочий стол для безнадёжных случаев.

Ладно, мне без разницы: я работаю в KDE и не намерен уходить на другой рабочий стол только потому, что вы назвали его крутым.
А никто и не заставляет. Но вы столкнётесь с ним в любом случае. Функция Activities Picker («Выбор задач») с её крупными значками адаптирована для сенсорных экранов. Функция Activities Overview [Обзор задач] рассчитана на доступ по нажатию единственной клавиши (по умолчанию это клави-



ша Windows), кнопки Activities на панели, в верхнем правом углу рабочего стола, или аппаратной кнопки. Видите, всё подготовлено для недорогого Linux-планшета, который вы собираетесь приобрести сразу после того, как он поступит в продажу. *Gnome Shell* может поджидать вас на любом мобильном устройстве, даже если вы не поклонник Gnome.

Ожидается ли на рабочем столе Gnome революция в стиле KDE 4.0?

Надеемся, что нет. Как мы уже говорили, Gnome весьма медлителен – в данном случае это здорово. Появляются новые оконные менеджеры – например, *Metacity* уступил место *Mutter* (который основан на инструментарии *Clutter* с поддержкой аппаратного ускорения), но более значительных перемен не предвидится. Приложения Gnome 2 будут работать на *Gnome Shell* без корректировки, хотя их разработчики безусловно портируют их на *Gnome Shell* (и Gnome 3), чтобы использовать все доступные функции. Переход на *Clutter/Mutter* может затруднить работу ПО для людей с ограниченными возможностями. Если вы пользуетесь подобными средствами при работе с компьютером, обратитесь по адресу https://webstats.gnome.org/Accessibility/GNOME3 и проверьте текущее состояние дел.

Стабильность, удобство использования… Готов поспорить, идея *Gnome Shell* нравится пользователям Ubuntu.

Угу, уж так нравится, что они предпочли ему *Unity*. Марк Шаттлворт как-то обмолвился: «Unity всё-таки оболочка для Gnome, хотя это и не *Gnome Shell*». Раскольники!

И где же можно подробнее узнать о *Gnome Shell*?

Без сомнения, самым лучшим будет установить его и попробовать. Если вы хотели бы ознакомиться с процессом разработки, загляните на сайт https://webstats.gnome.org/GnomeShell. Можно также отслеживать обновления своего дистрибутива: стабильная версия *Gnome Shell* ожидается в марте, а выход полноценной среды рабочего стола Gnome 3 запланирован на апрель. Более подробные сведения, рассчитанные не только на специалистов, можно получить на официальном сайте Gnome 3 – http://live.gnome.org/ThreePointZero. LXF



По рецептам доктора Брауна

Д-р Крис Браун
Доктор обучает, пишет и консультирует по Linux. Ученая степень по физике элементарных частиц ему в этом совсем не помогает.

Эзотерическое системное администрирование из причудливых заворотов кишок серверной

Настал год пингвина. Правда

Последние 10 лет я год за годом уверенно заявлял: «В этом году Linux, наконец, ждет взлет». Но решающих сдвигов так и не происходило – никто не сбрасывал статую Microsoft с пьедестала у подножия Борцов за Свободу Открытого ПО. И ежегодно я повторяю: «Настал год…»

Возможно, мне стоит переопределить термин «взлет». Если на популярные гаджеты вроде Kindle от Amazon или Bravia TV от Sony прорвался Linux – наверное, это взлет? Если правительство России объявило, что перейдет на свободное ПО в ближайшие четыре года – наверное, это взлет? (На русском языке можно прочесть об этом по ссылке http://open.cnews.ru/news/top/index.shtml?2010/12/27/421556,). Кстати, какой это владелец крупной фирмы-поставщика Linux свободно говорит по-русски?

Подозреваю, что по числу запущенных экземпляров ядра (если бы их можно было сосчитать) Linux уже доминирует в мире. Однако большинство этих экземпляров работает в потребительских устройствах, пользователи которых не в курсе, что такое ядро и тем более кто его написал, так же как не знают, что в гаджете есть процессор, и им не интересно, кто его изготовил.

Так или иначе, в 2011 году Linux, наконец, ждет взлет…

Снова о Тиме Тоуди

[Принцип TMTOWTDI, или "There's More Than One Way To Do It" – «Есть больше одного способа сделать это» (произносится «Тим Тоуди») – девиз языка Perl, — *прим. пер.*]

В **LXF140** я объявил конкурс на программу вывода строк файла в обратном порядке. И Давиде Бринде [Davide Brinde] прислал их 14 штук – от безупречных до нелепых. На все у нас места нет, но вот две однострочных:

```
cat -n infile | sort -k1,1m | cut -f 2-
```

и

```
sed '1!G;h;$!d' file
```

Спасибо, Давиде! У тебя явно избыток досуга.

Клуб любителей ядра

Ядро Думаете, ядро написал Линус Торвальдс? Это только 0,4 % истории. Узнайте, кто еще.

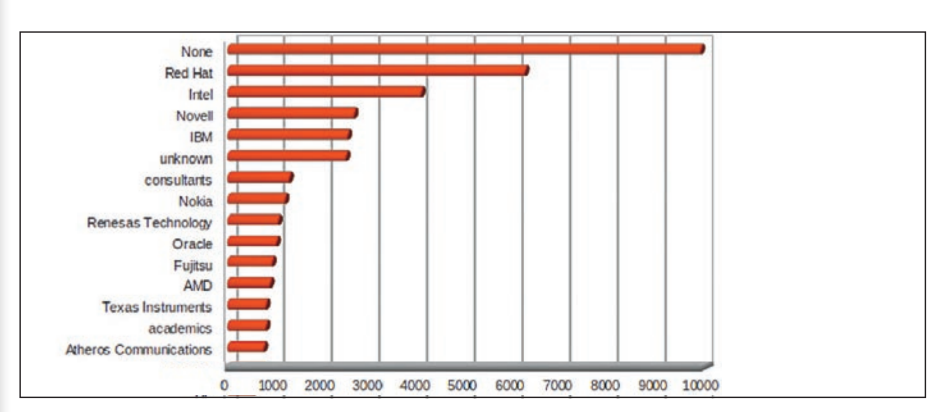
В декабре 2010 г. Linux Foundation опубликовал подробный доклад по разработке ядра Linux. Составленный в основном по материалам сайта kernel.org и репозитория ядра *Git*, он представляет любопытную оценку размера, организации и впечатляющего разнообразия сообщества. Немного статистики:

- Средний интервал между релизами – 81 день.
- Среднее количество заплат, применяемых к исходному коду ядра в час – 4,02.
- Число строк кода в ядре 2.6.11 в марте 2005 г. – 6624076.
- Число строк кода в ядре 2.6.35 в апреле 2010 г. – 13468253.

Также стоит отметить отсутствие «альфа-самца» в сообществе разработчиков. Список авторов изменений в ядре возглавил Дэвид Миллер [David Miller] (по числу изменений), но и он написал «всего» 1,3 % поправок. Более того, в тридцатку лучших не входит Линус Торвальдс, и, насколько я могу судить, женщин здесь тоже нет.

Распределив разработчиков по принадлежности к корпорациям, мы увидим другую картину. Во главе списка (19,1 %) – «вольные художники», не работающие ни на одну из корпораций. Первые 15 источников показаны внизу. В тридцатке лучших есть крупные игроки – Nokia, Oracle, Fujitsu, AMD, HP, Google, Samsung, но зияет отсутствие Canonical. Можете быть уверены, что никто из них не вносит свой вклад только по доброте душевной: дело в том, что их интерес к Linux как к компоненту их же ведущих коммерческих линеек растет.

Полная версия отчета – на http://bit.ly/gZpJLn. И это весьма поучительное чтение.



› **Кто спонсирует ядро Linux? Вот 15 основных «вкладчиков» по числу внесенных в код изменений с версии 2.6.30, представленных на горизонтальной оси.**

Удаленное управление

IPMI Доктор находит замену своей кочерге для управления сервером.

Раньше я полагал, что когда я выключаю свои серверы, они... кхе... выключаются. Но не тут-то было. Оказывается, маленькая микросхемка, контроллер управления базовой платой (BMC), продолжает работать – как тусклая свечка в окне сторожки у ворот замка, погруженного в ночную мглу. BMC – часть интеллектуального интерфейса управления платформой (IPMI), через который мы получаем информацию о состоянии устройств: температуру, напряжения, скорости вращения вентиляторов, открытости корпуса, состоянии питания и т.д. С помощью IPMI также можно получать информацию об устройствах, передавать предупреждения и даже удаленно включать и отключать питание сервера.

Изначально оговоренный Intel, HP, Nec и Dell, стандарт IPMI определяет общие интерфейсы управления системой, которые значительно облегчают управление гетерогенным набором серверов. Собственно стандарт – 644 страницы текста, не для слабых духом. Прочесть его можно по ссылке www.intel.com/design/servers/ipmi.

В архитектуре IPMI определяется устрашающее количество вариантов подключений к BMC. Есть «Шина управления интеллектуальной платформой» для подключения дочерних контроллеров, интерфейс для подключения к главной системной шине и подключения к датчикам и управляющим цепям, хранилище журнала событий внутри сеанса подключения и записей данных датчиков, а также внешние подключения к локальной сети через локальный порт и/или сетевую карту. Доступ к BMC через сетевое соединение, физически разделяемое с главным сетевым соединением компьютера – это удобное решение, не требующее дополнительных разъемов и кабелей, и этот подход используется в моем сервере Dell PowerEdge.

> «Linux, конечно, располагает утилитами для работы с IPMI.»

Linux, конечно, располагает утилитами для работы с IPMI. Они могут отслеживать и управлять системой на локальном компьютере, взаимодействуя с BMC через драйвер ядра и интерфейс системной шины. Есть также вариант «IPMI через локальную сеть». Я приведу примеры обоих подходов. И сразу же проясним: работа BMC совершенно не зависит от BIOS или любой локальной ОС. На самом деле, BMC будет работать даже при отключенном основном процессоре.

Чтобы опробовать локальное управление сервером, я установил Fedora 14 и пакет *ipmiutil*. В нем есть утилита *Ipmiutil* – по сути это командная обертка протокола IPMI, позволяющая отправлять IPMI-команды и получать результаты. Например, скорость вращения вентилятора можно узнать таким образом:

```
# ipmiutil sensor -c -g fan
ipmiutil ver 2.68
isensor: version 2.68
-- BMC version 1.30, IPMI version 2.0
supports device sdrs
Full sensor [000f] | snum 30 | FAN 1 RPM | OK* | 2280.00 RPM
isensor completed successfully
```

Для приведения в действие «IPMI через локальную сеть» на целевом компьютере нужно кое-что настроить. В BIOS я могу установить MAC-адрес и IP-адрес интерфейса LAN, которые не совпадают с сетевыми настройками компьютера. Там также можно задать имя пользователя, пароль и уровень привилегий, как объясняется во врезке «*Управление доступом*» вверху. Также обязательно включите «IPMI по локальной сети» [IPMI-over-LAN]. После этого вы сможете пообщаться с удаленным компьютером через IPMI, для чего я решил применить другой набор утилит (*ipmitool*) на удаленном компьютере с Ubuntu. На сей раз мне нужно указать IP-адрес удаленного компьютера, а также имя пользователя и пароль для аутентификации. В следующем примере я включаю сервер:

```
$ ipmitool -I lan -H 192.168.1.120 -U root -P mysecret chassis power on
Chassis Power Control: Up/On
```

Наконец, можно также прочитать журнал событий датчика, который лежит во постоянном хранилище в BMC, командой:

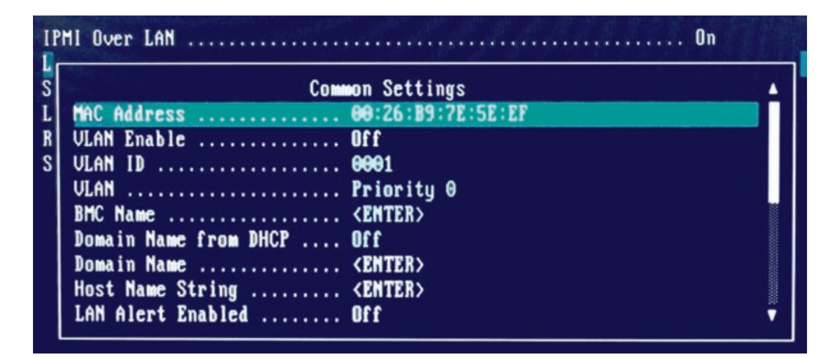
```
$ ipmitool -I lan -H 192.168.1.120 -U root -P farkle sel list
1 | 03/03/2010 | 14:06:32 | Event Logging Disabled #0x72 | Log area reset/cleared | Asserted
2 | Pre-Init Time-stamp | Physical Security #0x73 | General Chassis intrusion | Asserted
3 | Pre-Init Time-stamp | Physical Security #0x73 | General Chassis intrusion | Deasserted
```

»

Управление доступом

Вы можете поинтересоваться, как обстоит дело с безопасностью. Если какой-нибудь болван через Интернет узнает температуру моего процессора, горе тут невелико, но давать ему возможность удаленно выключить сервер – это уж будет квалифицироваться как атака типа отказ сервиса [DoS].

К счастью, протокол IPMI позволяет ввести в сеанс IPMI принудительную аутентификацию. Поддерживается несколько типов аутентификации, включая ее отсутствие [None] и аутентификацию с паролем [Straight Password], а также методы с шифрованием MD2 и MD5. Сам IPMI-трафик тоже можно зашифровать, и это возлагает большую вычислительную работу на скромный процессор BMC. Вдобавок пользователям назначаются уровни привилегий – Ответ [Callback], Пользователь [User], Оператор [Operator] и Администратор [Administrator]. Каждая команда IPMI требует определенного уровня привилегий. В общих чертах, Пользователи могут только читать информацию о состоянии системы. Операторы могут менять настройки типа порогов срабатывания датчиков и управлять системой (включать и отключать питание). Администратор может также настраивать сам IPMI, например, задавать имена пользователей и пароли. По сути это жесткая схема управления доступом на основе ролей.



› **Через экран BIOS сервера можно включить и настроить «IPMI по локальной сети», как описано выше.**

Реформы процесса загрузки

Systemd Пока борются титаны загрузки, доктор изучает отважного новичка.

В старые добрые времена процесс загрузки Unix BSD был доступен пониманию. Систему можно было запустить в однопользовательском режиме, и вы попадали в консоль пользователя root, или в многопользовательском режиме. Затем появилась *System V* с ее уровнями выполнения – одно из первых заболеваний типа «а добавим-ка на один уровень сложности больше, чем нужно». Тем не менее, люди привыкли к причудам файла **inittab**, символическим ссылкам **S*** и **K*** и перечню зависимостей в виде комментариев в скриптах запуска системы.

Затем появился *Upstart*. Я уже брюзжал, что происходящее в этой управляемой событиями системе трудно понять (**LXF129**). А теперь все изменится благодаря *systemd*, детищу умов Леннарта Поттеринга [Lennart Poettering] и Кая Сиверса [Kay Sievers]. Поясняю: это *systemd*, а не System D. Не было никаких систем A, B и C; это системный демон.

Цитирую страницу проекта: «*Systemd* – это системный менеджер и менеджер сервисов для Linux, совместимый с *SysV* и скриптами запуска LSB». *Systemd* предоставляет интенсивные возможности распараллеливания, запускает сервисы через активацию D-BUS и сокеты и демонов по требованию, отслеживает процессы с помощью cgroups, позволяет делать снимки состояния системы и восстанавливать по ним это состояние, поддерживает точки монтирования и автомонтирования и реализует тщательно проработанную транзакционную логику управления сервисами на основе зависимостей.

Раздувание процессами

В своем блоге (http://0pointer.de/blog) Леннарт приводит убедительный довод против «заражения процесса загрузки скриптами», указывая, что традиционные скрипты запуска *System V* часто вызывают *awk*, *sed* или *grep* для выполнения относительно простых операций. И каждый такой вызов означает создание нового процесса. Попробуйте сразу после загрузки открыть терминал и набрать следующую команду:

```
echo $$
```

Она даст вам представление о количестве процессов, перемалываемых системой при загрузке. У меня результаты были такими: в Fedora 14 и Ubuntu 10.10 (обе системы применяют *Upstart*) обнаружилось 1772 и 1611 процессов соответственно; в Red Hat Enterprise Linux (RHEL) 5 с традиционной *SysV* – аж 6781. *Systemd* может исключить многие из этих скриптов и распараллелить процесс загрузки, существенно ее ускорив.

Systemd изначально предназначалась для Fedora 14, но затем было решено отложить ее выход до Fedora 15, где она будет использоваться по умолчанию. Желая набраться опыта работы с *systemd*, я установил Rawhide, свежую рабочую версию Fedora. Так я получил доступ к рабочей версии системы и связанным с ней командам и man-страницам.

Главная утилита администрирования – *systemctl*. Без параметров она выводит список активных модулей, которыми управляет *systemd*:

```
$ systemctl
UNIT                          LOAD      ACTIVE SUB
JOB DESCRIPTION
sys-devi...ty-tty2.device     loaded    active plugged
/sys/devices/virtual/tty/tty2
dev-mqueue.automount          loaded    active running
POSIX Message Queue File System Automount Point
home.mount                    loaded    active mounted
/home
udev.service                  loaded    active running
udev Kernel Device Manager
systemd-logger.socket         loaded    active running
Logging Socket
cryptsetup.target             loaded    active active
Encrypted Volumes
systemd-...es-clean.timer     loaded    active waiting
Daily Cleanup of Temporary Directories
```

Я сильно сократил этот вывод, оставив по примеру для каждого из типов модулей, которыми управляет *systemd*. Они включают:

device	Устройство в дереве устройств Linux
automount	Точка автомонтирования файловой системы. У каждого модуля есть соответствующий модуль **mount**.
mount	Точка монтирования файловой системы.
service	Демоны, которых можно запускать, останавливать, перезапускать, перегружать.
socket	Сокет интернет-домена или Unix-домена. У каждого модуля есть соответствующий модуль **service**.
target	Логическая группа модулей (цель), аналогичная «метапакету» в управлении пакетами.
timer	Модули этого типа активируют другие модули по таймеру.
snapshot	Цель, которая сохраняет текущее состояние сервисов, чтобы позже его было можно восстановить.
path	Активация других модулей на основе существования файла или создание файла.

Более подробную информацию о модуле можно получить таким образом:

```
$ systemctl status udev.service
udev.service - udev Kernel Device Manager
          Loaded: loaded (/lib/systemd/system/udev.service)
          Active: active (running) since Tue, 04 Jan 2011 09:00:48
-0500; 6h ago
          Process: 459 (/sbin/udevadm trigger --type=devices
--action=add, code=exited, status=0/SUCCESS)
          Process: 443 (/sbin/udevadm trigger --type=subsystems
--action=add, code=exited, status=0/SUCCESS)
          Main PID: 410 (udevd)
          CGroup: name=systemd:/system/udev.service
                  ├ 410 /sbin/udevd
                  ├ 2160 /sbin/udevd
                  └ 2162 /sbin/udevd
```

Где узнать больше

Бурное обсуждение обоснования архитектуры *systemd* ведется в блоге Леннарта. (Начните с http://0pointer.de/blog/projects/systemd-for-admins-1.html). Man-страницы доступны на http://0pointer.de/public/systemd-man, они очень подробны и хорошо написаны. В данный момент поработать с *systemd* вживую непросто. Попробуйте установить Rawhide (инструкции по установке см. на http://fedoraproject.org/wiki/Releases/Rawhide), но помните, что релиз тестирован не до конца, и не найдетесь, что дорога будет торной.

Здесь есть масса информации. Указан файл настройки модуля (**udev.service**); а поместив процесс в группу (cgroup) **systemd:/system/udev.service**, *systemd* отслеживает не только родительский процесс демона (**410**), но и два дочерних (**2160** и **2162**). Мы также видим два других процесса, запущенных вместе с *udev* (**459** и **443**), и даже статусы их завершения.

Сервисы и группы

В традиционной системе Linux понять, какие процессы принадлежат данному сервису, может быть непросто. На имя исполняемого файла можно положиться не всегда, а связь с родительским процессом демоны часто разрывают путем «двойного ветвления». Однако дочерние процессы наследуют группу от родителя, а процессы, выполняющиеся не от имени root, не могут покинуть свою группу cgroup. Помещая сервис в группу, названную по имени сервиса, *systemd* получает гарантированный способ определения всех процессов, относящихся к этому сервису. Например, сигнал всем процессам, относящимся к сервису crond (группа **crond.service**) можно отправить командой

```
# systemctl kill crond.service
```

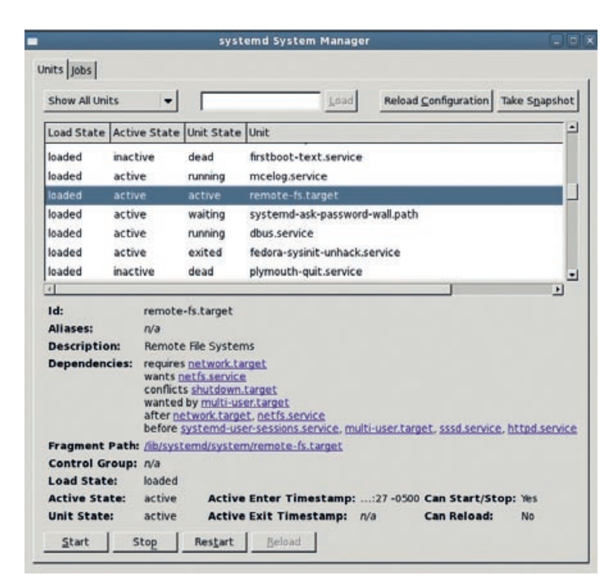
В пакете *systemd* есть утилита *systemd-cgls*, которая выводит список групп. Без параметров она выводит список групп, управляемых *systemd*. Приведенный ниже список сокращен.

```
$ systemd-cgls
 └ system
 ├ 1 sbin.init
 ├ sshd.service
 │ └ 2276 /usr/sbin/sshd
 ├ auditd.service
 │ ├ 874 auditd
 │ ├ 876 /sbin/audispd
 │ └ 877 /usr/sbin/sedispatch
```

В репозитории Rawhide также есть пакет *systemd-gtk* с графической утилитой *systemadm* для просмотра и управления модулями *systemd*. В ней можно получить информацию о состоянии модулей и их зависимостях и даже остановить и запустить модули.

Настройка

Настройки модулей содержатся в текстовых файлах, расположенных в каталогах **/lib/systemd/system** и **/etc/systemd/system**. Первый – сфера деятельности менеджеров пакетов, второй предназначен для локальных администраторов. Файлы с конфигурацией сервисов имеют расширение SERVICE, точек монтирования – MOUNT, и т. д. В сущности, файлы SERVICE – это замена загрузочным скриптам *System V*, но *systemd* создает модули и из обычных скриптов *Sys V*. Аналогично, файлы MOUNT служат заменой традиционному **/etc/fstab**, задавая используемые точки монтирования и опции монтирования. *Systemd* предоставляет обратную совместимость, создавая точки монтирования и из записей старого доброго *fstab*.



› **В противовес утилитам вроде *systemctl*, *systemadm* предоставляет графический интерфейс управления сервисами.**

Преимущества для всех

Когда я занимался разработкой научного ПО, то хорошо знал: чтобы ускорить выполнение программы, нужно проанализировать внутренние вложенные циклы в коде, которые выполняются миллионы раз. С этой точки зрения, процесс загрузки – последнее, что стоит оптимизировать. Однако для нетбуков, планшетов и других встроенных устройств потенциальное сокращение с *systemd* времени загрузки на несколько секунд – большой плюс. Для серверов, время холодной загрузки которых часто меньше, чем таймауты в BIOS, это не такая большая проблема. Тем не менее, улучшение управляемости и масштабируемость, вносимые *systemd*, должны вызвать улыбку и на суровых лицах сисадминов. Леннарт также намекнул, что на подходе – поддержка управления кластерами.

Рассмотрим пример файла настройки – в данном случае, это **/lib/systemd/system/avahi-daemon.service**. Номера строк даны для удобства ссылок.

```
1. [Unit]
2. Description=Avahi mDNS/DNS-SD Stack
3. Requires=avahi-daemon.socket
4. After=syslog.target
5.
6. [Service]
7. Type=dbus
8. BusName=org.freedesktop.Avahi
9. ExecStart=/usr/sbin/avahi-daemon -s
10. ExecReload=/usr/sbin/avahi-daemon –r
11. NotifyAccess=main
12.
13. [Install]
14. WantedBy=multi-user.target
15. Also= avahi-daemon.socket
16. Alias=dbus-org.freedesktop.Avahi.service
```

Разберем некоторые строки более подробно. Строка 3 задает зависимости: при активации модуля также будут активированы перечисленные в ней модули. Так как все большее количество сервисов «гуляют сами по себе», явно задается все меньшее количество зависимостей. Строка 4 определяет порядок запуска; она велит *systemd* запускать сервис *Avahi* только после **syslog.target**. Обратите внимание, что зависимость и порядок запуска определяются независимо друг от друга. Можно сказать, что A требует B, но если не указать явно, что A следует запускать после B, они запустятся одновременно. Наконец, строка 9 определяет команду, выполняемую при запуске сервиса.

На смену традиционному **/etc/inittab** – первому источнику информации о событиях во время загрузки – приходит модуль **default.target**. В Rawhide ему соответствует файл **/etc/systemd/system/default.target**, символическая ссылка на **/lib/systemd/system/graphical.target**. Это в какой-то степени соответствует определению уровня выполнения по умолчанию. Отсюда цепочка зависимостей, заданная в файлах настройки модулей, с помощью ключевых слов **Requires** и **Works** вызывает необходимые цели, запускает сервисы и монтирует файловые системы.

Отмечу, что миграция на *systemd* – дело непростое. Придется приложить немало усилий, и Леннарт в своем блоге и на man-страницах постарался объяснить, зачем он разработал эту систему. Но она идет по пятам *Upstart*, и я уверен, что не всем будет охота связываться с изучением нового механизма управления сервисами. LXF

LINUX FORMAT

Учебники

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Упомянул *PostgreSQL* на защите кандидатской диссертации по физике элементарных частиц.

Шрифты на миллион

Большое дело всегда требует
больших жертв
Это – равноценный обмен.
Ледяной алхимик

От шрифтов никуда не уйти. Вы постоянно пялитесь на текст, даже сейчас. Хорошие шрифты стоят либо дорого, либо совершенно неподъёмно. Чтобы повторить разработку шрифтовой коллекции Computer Modern с сопутствующей ей математикой, нужны десятки лет слаженных усилий сообщества. А разнообразия хочется прямо здесь, и не в отдалённом будущем.

Альтернатива, оказывается, есть: платить. Цена шрифта –миллион. Ком пания ПараТайп выпустила шрифты PT Sans (рубленый без засечек) и PT Serif (с засечками) под открытой лицензией; планируется и моноширинная версия. В эти шрифты включены все символы из алфавитов титульных языков РФ. Платило за них государство в лице Федерального агентства Роспечать.

PT Sans и PT Serif потихоньку расползаются как по Web'у, для которого они и создавались, так и по дистрибутивам GNU/Linux и/или LaTeX. Немного напрягает, что родительская фирма позиционирует их по сути как рекламу, параллельно предлагая более продвинутый коммерческий вариант. То есть расширяться имеющийся функционал свободных шрифтов не будет: тогда упадет ценность коммерческого набора. Например, в свободных шрифтах не нашлось места для болгарской кириллицы. Правда, цена этой проблемы известна – миллион; или придётся напрячься самому сообществу, ведь шрифты *уже* свободны.

E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Любить Gnome 52

Вам нравится Linux, и вы хотите извлечь максимум из этой настольной системы? Экскурсию по Gnome, рабочему столу по умолчанию в Ubuntu и Fedora, проводит для вас **Боб Мосс**.



Делать мультизагрузку 56

Множество людей держат двойную загрузку Linux и Windows, а как насчет Linux и Linux? **Эндрю Грегори** строит идеальный вариант для летунов с дистрибутива на дистрибутив.



Рулить проектами 60

Пусть название *Todoyu* кажется вам дурацким – сама программа вовсе не глупа и занимает достойное место в мире управления проектами. **Шашанк Шарма** пояснит принципы ее работы.



Исследовать FS 64

В среднестатистической установке Linux сотни каталогов; и чем же они занимаются? На Тренингах Linux от **Майка Сондерса** вы все поймете. Эх да ого-го!



Сортировать фотки 68

Фотографировать любят все, а вот возиться с переименованием файлов типа PC03133.JPG охотников мало. **Марко Фиоретти** и *ExifTools* помогут этому горю.



Создавать виджеты 72

Андрей Боровский рисует красивый индикатор с пружинной стрелкой, прикрытой колпачком, в стиле фотореализма. И все-таки она (стрелка) вертится.



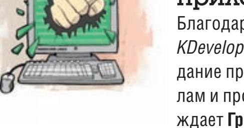
Менять платформу 76

Как перенести код C++ на платформу Mono? Ведь язык C# не поддерживает множественное наследование... Придется **Андрею Кузьменко** изучать подходы к реализации такового.



Читать на Erlang 80

Третий **Андрей**, **Ушаков**, показал, что функциональное программирование – не просто академическая дисциплина, а язык Erlang никак нельзя причислить к мертвым.

Лепить приложения KDE ... 84

Благодаря новоиспеченным KDE 4.6, *KDevelop 4.2* и *KAppTemplate*, создание приложений для KDE по силам и простым смертным, утверждает **Грэм Моррисон**.

Совет месяца: Webmin

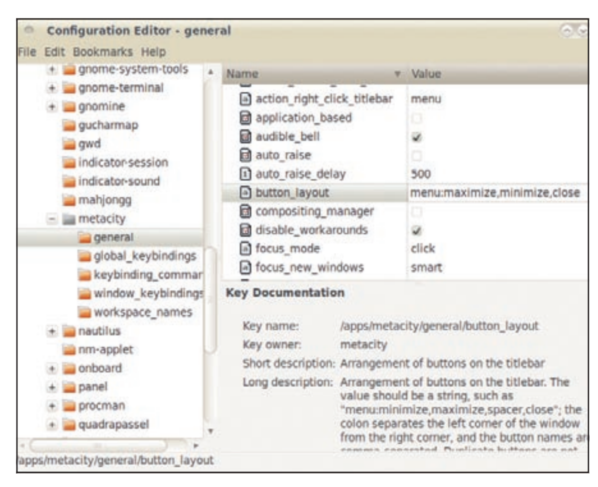
Текстовые файлы настройки – отличная штука: зная, что и в каком файле менять, вы гораздо быстрее добьетесь цели, чем продираясь через иерархию уровней GUI. Большинство дистрибутивов даже поставляют всякие программы настройки GUI, чтобы облегчить жизнь не доросшим до звания гуру. Но эти программы обычно годятся только для одного дистрибутива, требуют запуска рабочего стола и допускают лишь к тем опциям, которые соизволил разрешить автор. А если надо настроить удаленный сервер или сделать в вашем дистрибутиве такое, на что инструментов не предусмотрено?

Ответ на эту загадку – Webmin. Этот инструмент настройки на базе браузера написан на Perl и включает собственный web-сервер, так что будет работать на любой системе без заморочек с установкой *Apache*. После установки ваш дистрибутив обязан включить его в свои репозитории – укажите браузеру на **https://localhost:10000**, и увидите, что можно делать. Если вам нужен доступ с другого компьютера, но по умолчанию ваш дистрибутив блокирует соединения извне, запустите его локально, используя ссылки по SSH, как на безголовом сервере, и настройте параметры в разделе *Webmin*.

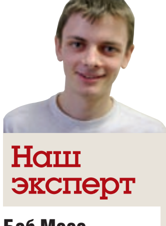
Gnome: Новое

Если вы лишь недавно вызволились из клещей другой ОС, следуйте за **Бобом Моссом**: он представит вам рабочий стол Gnome.





› **Можно поменять настройки по умолчанию в Gnome с помощью gconf-editor.**



Наш эксперт

Боб Мосс
Разрывается между написанием кода и изучением компьютерных наук.



› **Использование нескольких рабочих столов избавит вас от бесконечного нажатия Alt+Tab для переключения между окнами.**

В прошлом месяце мы вышли за пределы области первых шагов и расширили наши знания о свободном офисном пакете *OpenOffice.org*. На сей раз мы возвратимся к основам и пристально рассмотрим рабочий стол Gnome. Он часто принимается как должное, но эта платформа поддерживается практически всеми дистрибутивами, независимо от того, основной это рабочий стол или нет. Более того, благодаря успеху дистрибутивов вроде Fedora и Ubuntu, Gnome по-прежнему остается средой, которая первой встречаает новых пользователей.

Однако Gnome далек от совершенства. Несмотря на пуленепробиваемую надежность, есть еще изъяны, способные смутить неосмотрительных пользователей Mac или Windows при их попытке разок поработать со свободной операционной системой. Здесь мы рассмотрим все, что нужно знать для начала работы, и как настроить Gnome в соответствии с вашими потребностями, и какие перспективы развития ждут эту среду.

Потихоньку приступим

В новой рабочей среде Gnome вам прежде всего нужно знать, где найти важные приложения, используемые каждый день. Первое из них – *Mozilla Firefox*, и, как правило, оно идет в комплекте с большинством дистрибутивов Gnome. В Ubuntu по умолчанию вы увидите его иконку рядом с меню Система в верхней панели, или в меню Приложения > Интернет. Для обработки текстов и прочего есть *OpenOffice.org*, обычно находящийся в Приложения > Office (если его нет, просто следуйте инструкциям в окне *Установка программ*).

Вы уже наверняка поняли схему действий. Все приложения хранятся в меню Приложения Gnome и четко классифицированы в зависимости от функции. Например, окно терминала находится в Приложения > Стандартные, а фотоменеджер – в Приложения > Графика. Возможно, вам это покажется не похожим на беспорядочное меню Пуск в Windows или dock в OS X, но привыкнув к нему, вы почувствуете, что благодаря этой классификации поиск приложений сильно ускоряется.

Кстати о скорости

Чтобы открыть файлы, перейдите в меню Места через раскрывающийся список наиболее часто используемых папок. Здесь вы увидите ваши документы, музыку, видео и фотографии, а также все внешние жесткие диски, USB-карты памяти или SD карты, подключенные к системе.

Если вы хотите настроить вашу систему, большинство опций для этого находятся в меню Система. Например, чтобы добавить пользователя, перейдите в Система > Администрирование > Пользователи и группы; или можете настроить вашу батарею и параметры питания, перейдя в Система > Настройки > Управление питанием.

Вы будете рады узнать, что толковая организация приложений, файлов и настроек системы – не единственное достоинство рабочего стола Gnome. Можно также использовать нечто, известное как множественные рабочие столы – в среде Windows вы, возможно, с таким не встречались. В случаях, когда открыто несколько окон, часто удобнее организовать их более чем на одном экране. Безусловно, при использовании Alt+Tab достаточно легко переключаться между четырьмя-пятью окнами, но когда число окон приближается к двузначным цифрам, это становится

» **Месяц назад** Мы расширили макросами функционал *OpenOffice.org*.

начало



› **Установка программ через *Ubuntu Software Centre* невероятно проста. См. также: Linux Mint и дистрибутивы KDE, применяющие *PackageKit*.**

неподъемным. В мире Linux нажатия Ctrl+Alt и правой кнопки достаточно, чтобы переключиться на другой рабочий стол, и Ctrl+Alt+Влево, чтобы вернуться. Для перемещения окна между рабочими столами просто щелкните правой кнопкой мыши на его заголовке и тащите его оттуда в рабочую область слева, справа или где попало.

Виртуальное безумство

По умолчанию большинство дистрибутивов для начала предлагают только два виртуальных рабочих стола, но если вы щелкнете правой кнопкой мыши на их иконках внизу справа на экране и нажмете Настройки, можно включить столько, сколько вашей душе угодно. Можно, к примеру, отвести один рабочий стол для работы в Интернет и обмена мгновенными сообщениями, другой – для редактирования файла, а третий – для проигрывания музыки. Короче говоря, виртуальные рабочие столы сделают вашу работу продуктивнее, позволив быстро переключаться между различными видами деятельности без необходимости сворачивать и разворачивать десятки окон.

Установка приложений

Установка пакетов в среде Gnome зависит от дистрибутива, точнее, от используемого в нем менеджера пакетов. В Fedora, например, программное обеспечение устанавливается через Система > Администрирование > Установка и удаление программ, или с помощью такой командной строки:

```
yum install gedit-plugins
```

Здесь **gedit-plugins** означает любой пакет программ, которые вы хотите установить. Пользователи Ubuntu должны выполнять другие шаги. Можно сделать так:

```
sudo apt-get install gedit-plugins
```

В отличие от Fedora, однако, не обязательно прибегать к терминалу. Здесь есть *Ubuntu Software Centre*, сильно упрощающий жизнь. Для запуска пакета перейдите в Приложения > Ubuntu Software Centre. Здесь вы увидите приложения, распределенные по категориям: можно увидеть те, что рекомендует устанавливать Canonical, или ввести термин в окне поиска и просмотреть список пакетов, обеспечивающих требуемую функциональность.

Для получения подробной информации о приложении дважды щелкните по нему. Если вы хотите установить его, просто нажмите кнопку Установить и введите свой пароль. Вы можете наблюдать за ходом процесса или добавить в очередь на установку другие пакеты. Аналогично, для удаления имеющегося ПО, нажмите на кнопку Удалить в том же месте, где вы нажимали для его установки.

Теперь, когда вы знаете, как настроить систему, запустить приложения и распределить окна по нескольким рабочим столам, почему бы не разобраться с воспроизведением музыки и видеофайлов. Для большинства медиа-файлов, который вы хотели бы запустить – например, с расширениями OGG, MP3 и AVI – вы увидите, что они открываются просто двойным щелчком по файлу и проигрываются в минимальном пакете по имени *Mplayer*. Однако мы хотим организовать и проигрывать их так же, как и на компьютере под Windows или Mac. По части музыки есть пакет под названием *Rhythmbox*, который часто идет в комплекте с Gnome-дистрибутивами. Его можно найти в Приложения > Звук и Видео.

После запуска приложения появится запрос на импорт музыкальной папки; тут-то вы и можете выбрать свою музыкальную коллекцию. Наконец, если необходимы дополнительные кодеки – скажем, для воспроизведения MP3-файлов – то большинство основных дистрибутивов предлагает установить их для вас тоже.

Скорая помощь

Чтобы настроить клавишу Windows на открытие меню Приложения Gnome, просто перейдите в Система > Настройки > Комбинации клавиш, нажмите Далее, чтобы показать главное меню Панели, и нажмите клавишу Windows.

Причуды кнопок Ubuntu

Если вы решили попробовать Ubuntu, позвольте нам выразить сожаление, что команда разработчиков решила во всех окнах поместить кнопки Свернуть, Развернуть и Закрыть вверху слева. Хотя пользователи Mac при этом быстрее освоятся в интерфейсе Ubuntu, остальные 95 % пользователей компьютеров, которые привыкли к Windows и видят Linux первый раз, будут разочарованы.

Но не отчаивайтесь: есть способ исправить эту особенность дисплея на что-то более знакомое, не прибегая к командной строке. Нажмите Alt+F2 и наберите:

```
gconf-editor
```

Теперь нажмите клавишу Enter для запуска команды. Вы должны увидеть новое окно с названием Configuration Editor.

Откройте древовидное отображение Службы > Metacity > Общие и прокрутите вниз панели справа, пока не найдете button_layout. Затем нужно будет дважды нажать на значение, чтобы изменить его на следующее:

```
menu:minimize,maximize,close
```

Учтите, что в строке полностью отсутствуют пробелы. Кроме того, имейте в виду, что если вы еще используете Ubuntu Lucid (10,04), это может привести к неадекватному рендерингу кнопок, и взамен рекомендуется использовать строку

```
menu:maximize,minimize,close
```

Нажав кнопку OK, вы обнаружите, что ваши изменения сразу же применились и на рабочем столе восстановлена привычная картина.

Gnome: Копнем

В Gnome есть масса интересного; давайте рассмотрим, как настроить ваш рабочий стол «под себя».

Теперь в основах вы разобрались, и настало время помочь Gnome отразить вашу индивидуальность, изменив его тему и добавив 3D-эффекты, чтобы изумлять друзей. Для начала просто щелкните правой кнопкой мыши на рабочем столе и выберите Изменить фон рабочего стола. В появляющейся панели большинство дистрибутивов предоставляет широкий спектр различных обоев, а Ubuntu также включает и пару динамических, которые периодически меняются. Перейдя затем на вкладку Темы, вы сможете изменить внешний вид всех окон и панелей. Наконец, если вы хотите добавить эффекты, перейдите к вкладке Эффекты и переключитесь между Базовым, Нормальным и Экстра.

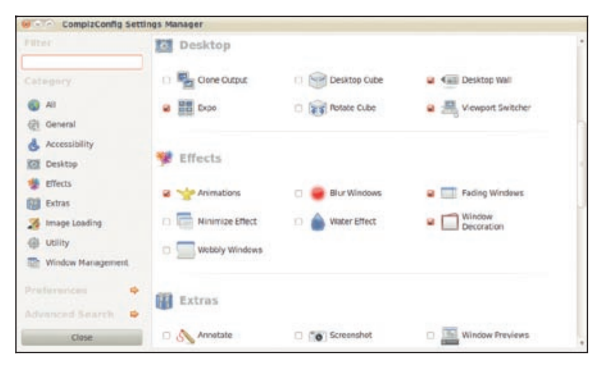
Впрочем, учтите вот что: возможно, часть этого будет заморожена, поскольку требует для своей работы несвободных драйверов, назло Ричарду Столлмену и ему подобным. Не паникуйте – денег доплачивать никому не нужно; но придется перейти в Система > Администрирование > Дополнительные драйверы или аналогично, и нормально активировать графические драйвера Nvidia или ATI. После перезагрузки это опции станут доступны.

Но раз уж вы обратились к Linux, это, вероятно, означает, что вы ищете намного большую гибкость. Вы будете рады узнать, что можно получить детальный контроль над 3D-графикой системы с помощью CCSM (*CompizConfig Settings Manager*) и *Emerald*, менеджера тем. Оба можно установить практически из любого основного менеджера пакетов.

Чтобы ваша система вращалась, как тот самый куб, при каждом перемещении между рабочими столами, откройте CCSM через Система > Настройки > Менеджер настройки CompizConfig. Перейдите к рабочему столу и поставьте галочки для Desktop Cube и Rotate Cube; нажмите кнопку Разрешить конфликты в любых окнах, которые появляются. Для пульсирующих окон, прокрутите Эффекты, а затем нажмите Wobbly Windows. Для еще более точного управления, жмите на соответствующие значки и настройки значений переменных на ваш вкус.

Скорая помощь

Если ваша система зависает, когда вы резвитесь с настройками, возможно, стоит перезапустить *X Server*: нажмите Ctrl+Alt+Backspace или (на системах, где это отключено) нажмите Ctrl+Alt+F1 и наберите **/etc/init.d/gdm restart**.



› *CCSM* поначалу кажется странным; но, поэкспериментировав с настройками, вы убедитесь в его мощи.

Чтобы воспользоваться темами *Emerald*, перейдите на www.compiz-themes.org и нажмите на ссылку Emerald в меню слева. Разархивируйте все загружаемые ZIP-файлы, а затем импортируйте файл EMERALD в *Emerald*. Просто нажмите на импортированную тему, и она применится к вашей системе.

Забегая вперед

Освоив Gnome 2, вы, возможно, захотите узнать, что заготовил Gnome 3. Команда Gnome сейчас работает над *Gnome Shell* (см. *Завернутый в оболочку*, ниже), в то время как Canonical работает над Unity для Ubuntu. Хотя они в стадии разработки, можно установить их оба и попробовать супер-новые интерфейсы, которые попадут в мир Linux в ближайшие месяцы.

Если вы планируете познакомиться с ними на виртуальной машине, возможно, вам не повезло, так как пакеты типа *Oracle*

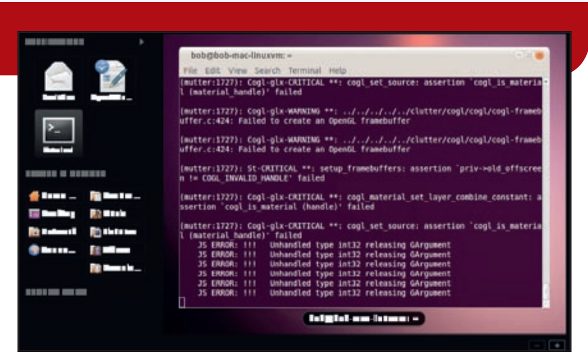
Завернутый в оболочку

Если Ubuntu не для вас, но вы все-таки интересуетесь, что предлагает Gnome 3, посмотрите *Gnome Shell*. Это интерфейс, используемый всеми остальными Gnome-дистрибутивами, хотя на момент написания он по-прежнему в стадии разработки и поэтому может показаться не самой простой программой для работы.

Gnome Shell доступна для установки в большинстве основных менеджеров пакетов. Для запуска, просто наберите в терминале

```
gnome-shell --replace
```

По умолчанию на экране вы не найдете много информации. Это потому, что вам нужно запустить приложения в вашей Activity (по сути, один из ваших многочисленных рабочих столов, под другим названием). Щелкните на Приложения в верхнем левом углу, и вы увидите предпросмотр рядом с появившимся меню слева. Отсюда можно запускать часто используемые приложения и недавно открытые документы или вводить термин в окне поиска. Если вам захочется добавить или удалить лишние действия или рабочие столы, просто нажмите на иконку плюс или минус в нижней правой части экрана, как раз под предпросмотром.



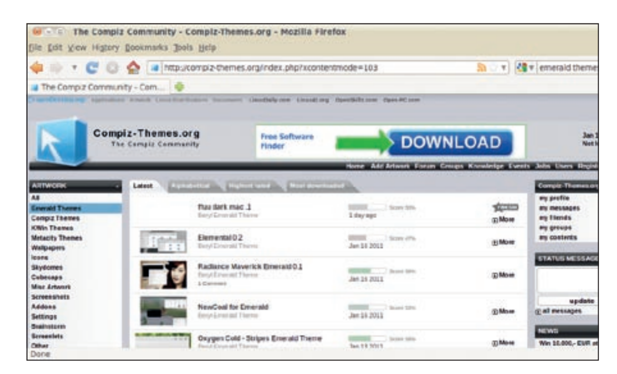
› Как видите, *Gnome Shell* все еще в стадии разработки на момент написания этой статьи!

» **Пропустили номер?** Узнайте на с. 107, как получить его прямо сейчас.

чуть глубже

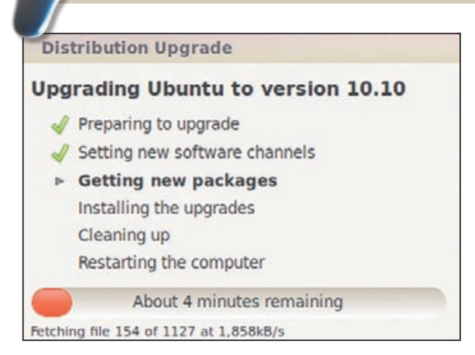
VirtualBox на момент написания непригодны. Хотя эти вопросы, возможно, будут разрешены, когда вы прочтете эти строки, все равно будьте осторожны, особенно в производственной среде.

В Unity вы найдете, что приложения, которые вы, скорее всего, будете использовать, расположены на левой панели – но это можно изменить, удалив или перетащив сюда другие приложения. При необходимости очистить рабочий стол просто жмите на значок Ubuntu. Вы также заметите, что между Unity и оболочками Gnome много общего. Однако есть тонкие различия, так что оценить ваш будущий рабочий стол можно только на деле. По этому вопросу в рамках всего общества Linux есть разные мнения, так что не стесняйтесь высказывать любые мысли на форумах **LXF**.



› Множество разнообразных, легкоустанавливаемых тем Emerald доступно для скачивания бесплатно на www.compiz-themes.org.

Шаг за шагом: Попробуйте Unity



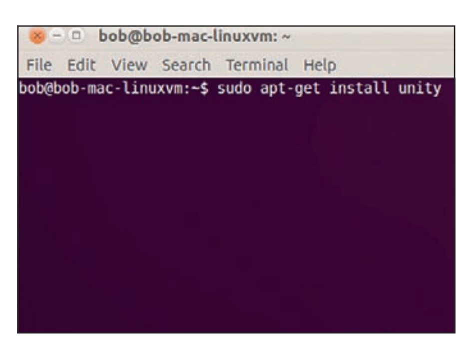
1 Обновление дистрибутива

Хотя обновление не обязательно, при запуске последней сборки Unity на более ранних версиях Ubuntu могут возникать проблемы.



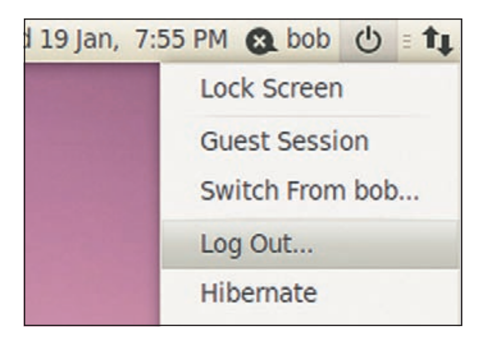
2 Подготовка к установке

Добавьте PPA при помощи команды, показанной выше, затем используйте обычное обновление для повторной загрузки списка пакетов.



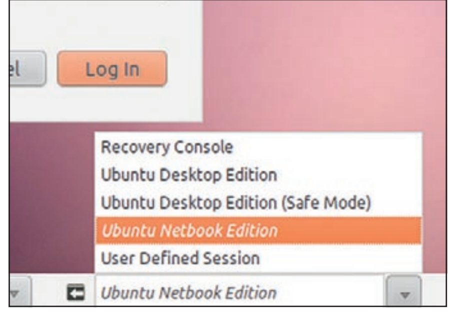
3 Установка Unity

Установка выполняется одной командой терминала (на момент написания в *Ubuntu Software Centre* это не было доступно).



4 Выход

Выйдите из сессии Gnome, чтобы сохранить текущие файлы и изменения, затем запустите *Gnome Display Manager*.



5 Вход и запуск

Для запуска Unity потребуется выйти, нажать на Сессии, затем Ubuntu Unity Network Edition, после чего ввести параметры логина.



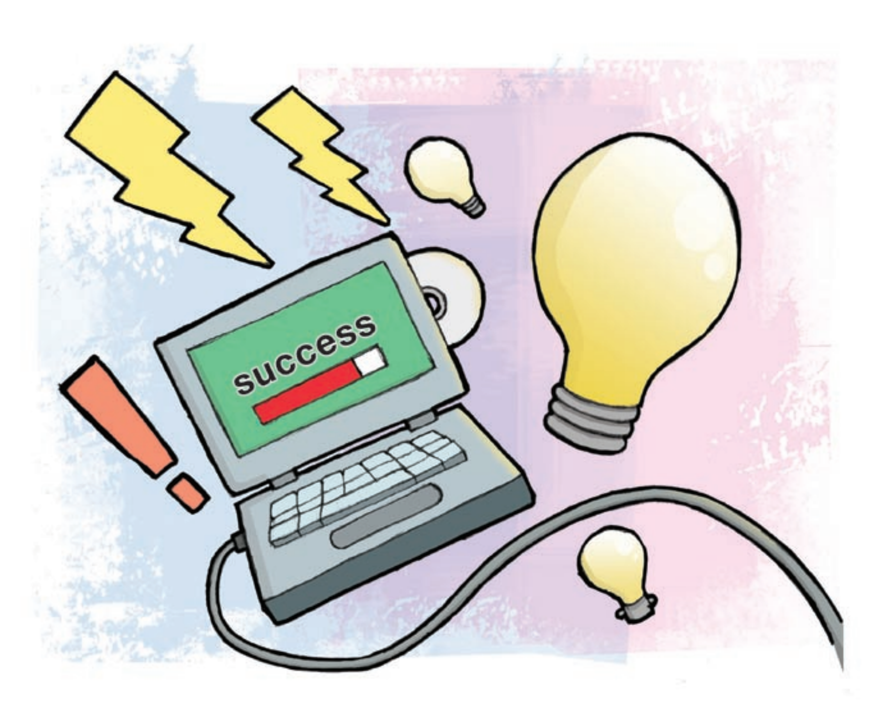
6 Запуск приложений

После входа, программы, которые вы, скорее всего, будете использовать, отобразятся на панели слева, как и возможности дополнительной загрузки. **LXF**

» Через месяц Мы слегка затронем KDE и приспособим его к нашим потребностям.

Grub: Дважды

Двойная загрузка хороша для запуска Linux вместе с Windows; ну, а если вам нужны два варианта Linux? За дело берется **Эндрю Грегори**.



Одна из прелестей Linux – то, что если вам надоест свой дистрибутив, вы можете безвозмездно, то есть даром, переключиться на другой: не нужно разоряться на установочные диски, вводить коды активации, доказывая, что вы тот самый человек, который заплатил за программы, или заботиться об установке драйверов третьих сторон. Жизнь прекрасна.

Единственное, что тут может не понравиться – это неприятность с резервированием и восстановлением тонн данных при каждом таком переходе. Если вы снуете, как белка, между Ubuntu и Fedora или какой-нибудь другой находкой с очередного **LXFDVD**, вы, чего доброго, потратите больше времени на перенос данных, чем Майк играет в *Frontier*.

Но вовсе не обязательно очищать жесткий диск при каждой смене дистрибутива Linux. Инсталлятор дистрибутива можно задействовать для создания на жестком диске отдельной области для данных и отдельной – для файлов вашего дистрибутива, и при переустановке дистрибутива данные не трогать. Как завзятые умники, мы установим систему с двойной загрузкой Ubuntu и Fedora, чтобы вы могли попробовать два светоча мира дистрибутивов, используя одни и те же файлы, и решить самостоятельно, который вам милее. А дальше решайте по своему усмотрению – можете вообще установить что-нибудь третье, а все данные будут на месте.

Часть 1: Установка Fedora



Наш эксперт

Эндрю Грегори пользуется Linux уже многие годы и обтесался на множестве руководств **LXF**; вот почему у него так блестят волосы.

Поскольку мы устанавливаем систему с двойной загрузкой, потребуется задействовать *Grub* (grand unified bootloader), а это боязно. Напортачив в файле настройки любого другого приложения Linux, можно просто открыть терминал (или нажать Ctrl+Alt+F1, если вы сломали графический дисплей), перейти туда, где находится порченый файл, и исправить его – либо отредактировав вручную место, вызывавшее ошибку, либо заменив весь файл ранее созданной резервной копией. *Grub* же отличается тем, что если он не работает, то не работает ничего. При поломке *Firefox* можно просто выйти в Интернет из *Konqueror*, *Chrome* или *Epiphany*, но сломайте *Grub* – и ваша машина перестанет загружаться, а это тяжелый случай. Поэтому лучше всего изучать тонкости *Grub* из виртуальной машины, храня данные в безопасности. Разные дистрибутивы общаются с *Grub* по-разному, однако Ubuntu настолько умен, что нам самим не придется редактировать файлы настройки. А если кому надо, подписчики найдут подробные руководства по *Grub* и *Grub 2* в **LXF127** и **LXF128**.

Что вам потребуется

- **1 ГБ ОЗУ** Наш старый ноутбук Dell с 512 ГБ ОЗУ не потянул виртуализацию одного полнофункционального рабочего стола внутри другого.
- **6 ГБ места на диске** Мы собираемся создать виртуальный жесткий диск размером 50 ГБ, что соответствует файлу размером примерно 5 ГБ. Дополнительный гигабайт добавлен для подстраховки.
- **VirtualBox** Доступен в репозиториях множества дистрибутивов или на сайте www.virtualbox.org.
- **ISO-образы Ubuntu и Fedora**.

К порядку!

Мы хотим, чтобы у нас была система с двойной загрузкой Fedora 14 и Ubuntu 10.10 – и вот первое решение, которое надо принять: с какого дистрибутива начнем? Хотя весь смысл *Grub* состоит в возможности загрузки нескольких ОС с одного жесткого диска, одни дистрибутивы подходят лучше, другие – хуже. По умолчанию установщик Fedora игнорирует другие системы Linux и дает только один вариант – загрузить Fedora, даже если у вас установлены другие ОС Linux. Поэтому сперва мы установим Fedora, а затем Ubuntu. Установщик от Canonical гораздо лучше распознает ранее установленные системы; он отыщет установку Fedora и позволит вам загружать оба этих дистрибутива.

Поместив Fedora в привод, загрузите рабочий стол live и дважды щелкните по значку Install To hard Drive. Задайте, как обычно, установки языка и клавиатуры. Возможно, вы увидите сообщение, предупреждающее, что жесткий диск нуждается в повторной инициализации. Можете спокойно принять опцию Reinitialize All, стоящую по умолчанию.

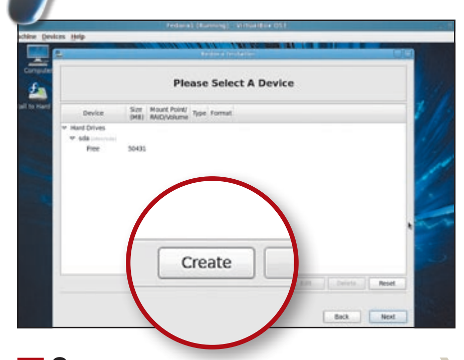
Продолжите установку, пока вам не встретится экран с вопросом о типе установки; выберите Create Custom Layout. Тут вы ока-

Linux

жетесь в совершенно неочевидной утилите Fedora для создания разделов диска; как с ней обращаться, подробно расписано в инструкции *Шаг за шагом* ниже. Жмите Next, игнорируя все выводимые предупреждения. Если, по-вашему, это бездумное гусарство, вспомните, что жесткий диск, на котором вы работаете, виртуальный и в случае ошибки худший вариант потери – только затраченное время.

Fedora далее показывает все операционные системы, появляющиеся при загрузке. Покамест это только Fedora. Вас спросят, хотите ли вы установить загрузчик на sda (оставьте все по умолчанию). Нажмите Next и включите чайник. Пока вы смакуете чашку чая, произойдет копирование файлов, и вы сможете задать имя пользователя и пароль, а также проверить настройки даты и времени.

Шаг за шагом: Разбиваем диск



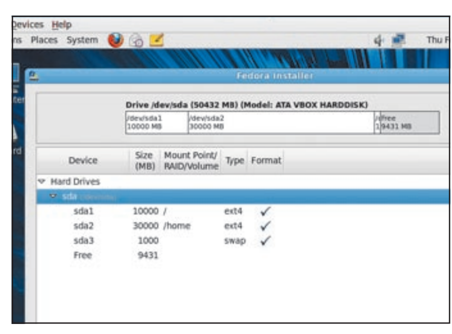
1 Создание раздела

Поскольку установка ведется на виртуальный жесткий диск, программа – создатель разделов покажет только одно устройство, с именем sda. Чтобы начать, нажмите кнопку Create; откроется другой диалог, где вас спросят про тип создаваемого раздела. Выберите Standard Partition и нажмите на Create еще раз.



2 Создание раздела root

В выпадающем меню Mount Point выберите / (косая черта). Это будет корневым разделом, куда Fedora установит свои файлы и где будут содержаться все установленные программы. Полная установка Fedora может превысить 9 ГБ, так что с учетом погрешности сделаем корневой раздел размером 10 ГБ.



3 Подкачка и /home

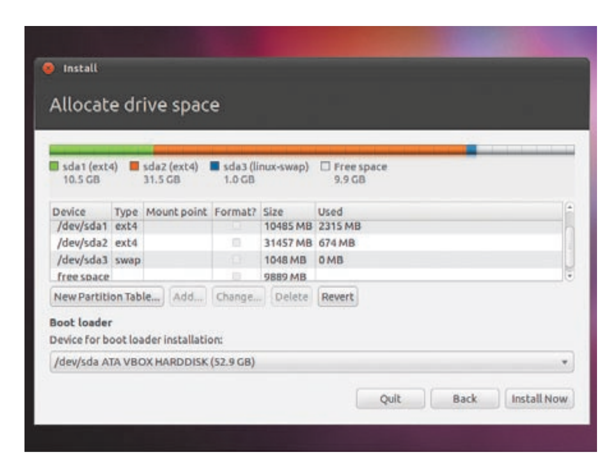
В выпадающем списке Filesystem Type выберите ext4, затем нажмите на OK. Нажмите на Create еще раз, чтобы создать раздел подкачки размером 1 ГБ (при выборе **swap** в списке файловых систем опция Mount Point станет неактивной) и 30-ГБ раздел типа ext4 с точкой монтирования **/home**. Оставшееся место пойдет на Ubuntu.

Часть 2: Установка Ubuntu

Итак, у вас есть установка Fedora 14, запущенная в виртуальной системе. Теперь установим нашу вторую ОС – популярную Ubuntu.

Поменяйте в меню *VirtualBox* Storage ISO-образ Fedora на ISO Ubuntu и перезапустите виртуальную машину, чтобы попасть в рабочий стол Ubuntu live. Начните обычный процесс установки; однако, дойдя до приглашения Allocate Drive Space, выберите Specify Partitions Manually (Advanced).

Нажмите на Free Space, расположенную внизу списка разделов, а потом на Add. Открывающийся диалог будет вам уже знаком: это тот же диалог, что был в программе Fedora, создающей разделы, только слегка визуально приглаженный. Как и ранее, создадим корневой (/) раздел для Ubuntu, используя файловую систему ext4. Отличие будет в том, что также необходимо поменять тип раздела с Logical на Primary. Нажав OK, вы вернетесь в обзор разделов, после чего выберите раздел **/home**, созданный в Fedora, и нажмите Change. Мы создали этот раздел для совмест-

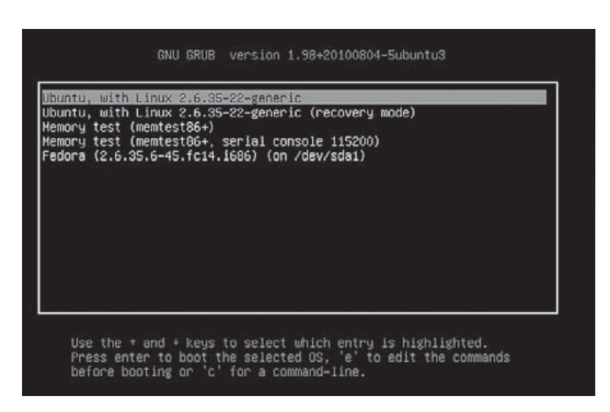


› Редактор разделов Ubuntu отображает разделы, созданные нами в Fedora, и свободное место, где мы установим Ubuntu.

ного использования в Fedora и Ubuntu, но пока установщик Ubuntu считает его просто областью на диске, где есть некие данные. Чтобы Ubuntu распознал его как раздел **/home**, его нужно примонтировать. Для этого поменяйте в выпадающем меню Use As опцию Do Not Use This Partition на опцию Ext4, и присвойте ему точку монтирования **/home**.

Вам осталось только установить язык, клавиатуру и свое местоположение и создать имя пользователя (обратите внимание, что его имя должно отличаться от того, что было в Fedora) с паролем. Затем, пока идет процесс копирования файлов, поставьте чайник для второй заварки.

После перезагрузки вы увидите экран *Grub*, где можно выбрать между Fedora и Ubuntu – как и было задумано! Пару секунд упивайтесь своим успехом, а потом мы пойдем дальше: системы надо подружить.



› Ура! У нас есть система с двойной загрузкой, а файлы раздела /home можно использовать в обеих ОС.

Часть 3: Разделение файлов между дистрибутивами

Установив две системы Linux, следует разработать способ для совместного использования этими системами одних и тех же данных. Вы, наверное, думали, что создание общего раздела **/home** облегчит задачу. Это так, но для полного комфорта нужно сделать еще несколько шагов.

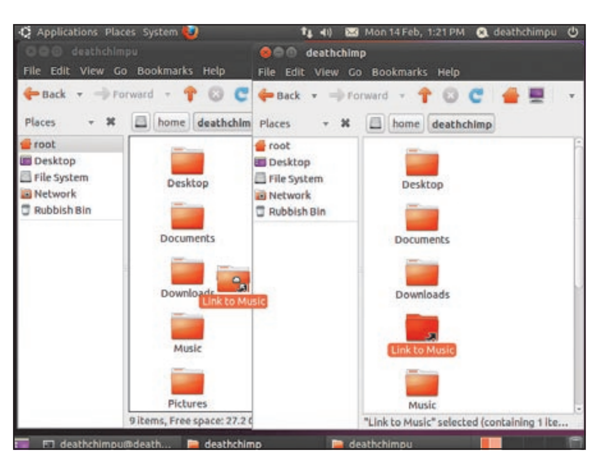
Вы ведь помните, что мы создали разные учетные записи пользователя для каждого установленного дистрибутива? И вот почему: наряду с папками **Documents**, **Music** и **Videos**, а также другими полезными для обеих систем файлами, ваш домашний каталог содержит некоторые папки, которыми им явно не следует делиться. Заглянув в домашний каталог, вы, возможно, не поймете, о чем это мы. Но если в *Nautilus* нажать Ctrl+H (что эквивалентно опции **-a** в команде **ls -a**, показывающей скрытые файлы), вы увидите папки, имя которых начинается с точки, при нормальных условиях работающей для этих папок как шапка-невидимка. Эти папки содержат файлы настройки, которые задают рабочий стол Gnome, а также пользовательские параметры. Если два ваших дистрибутива имеют две версии одной и той же программы, управляемые одним и тем же набором файлов настройки, это не может не привести к несовместимости, и необходимы два домашних каталога со своими файлами настройки для каждого дистрибутива. Правда, это приводит к дублированию и таких папок, как, например, **Music**, однако в домашнем каталоге дистрибутива Ubuntu мы будем использовать символические ссылки на фонотеку пользователя Fedora.

Что за UID?

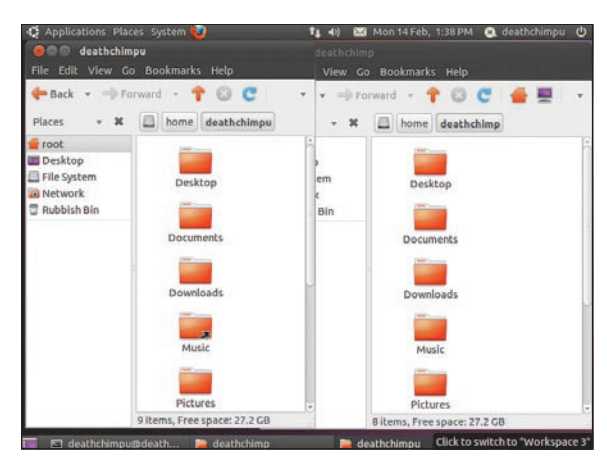
Linux управляет правами доступа к файлам через систему ID пользователя (UID) и групповых ID (GID). К сожалению, в Fedora и Ubuntu эта идея реализована немного по-разному: Fedora присваивает пользователю UID, равный 500, а Ubuntu (а также Debian, Mint, Slackware и многие другие) использует значение UID 1000. Различие не принципиальное, но приносит дополнительную головную боль. К счастью, есть довольно простой способ обойти его: создать новую группу с именем 'dualboot', которая будет иметь одинаковый групповой ID в обоих дистрибутивах. Затем можно сделать ее группой-владельцем всех совместно используемых папок.

Откройте окно Users And Groups из System > Administration. Щелкните по Groups, затем Add для создания новой группы. В возникающем окне задайте группе имя 'dualboot', присвойте ее GID равным 600 и поставьте галочку, чтобы ваш пользователь Ubuntu стал членом этой новой группы.

Создав группу, можно поменять права доступа всех каталогов, предназначенных для совместного использования двумя дистрибутивами, чтобы эти каталоги были доступны для членов группы. Например, щелкните правой кнопкой по папке **Music** и выберите в меню пункт Properties. Перейдите на вкладку разрешений и смените параметр Group на dualboot; при этом Folder Access должен быть Create And Delete Files. Щелкните на Apply Permissions To Enclosed files и закройте окно.

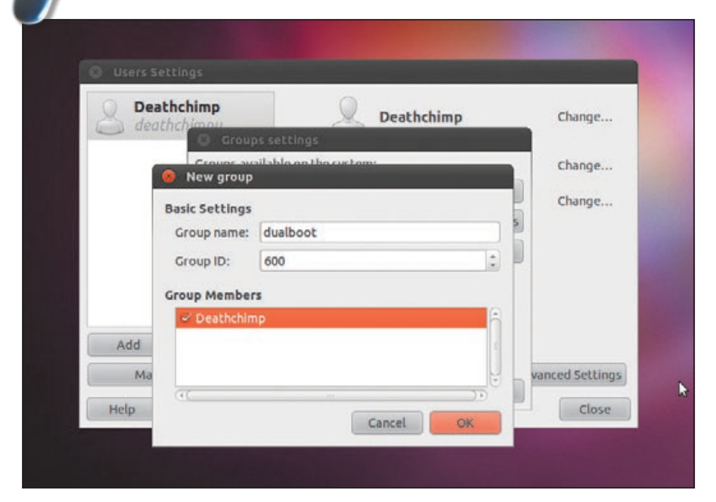


› Создайте ссылку в меню по правой кнопке, а затем перетащите ее в другой пользовательский каталог home.



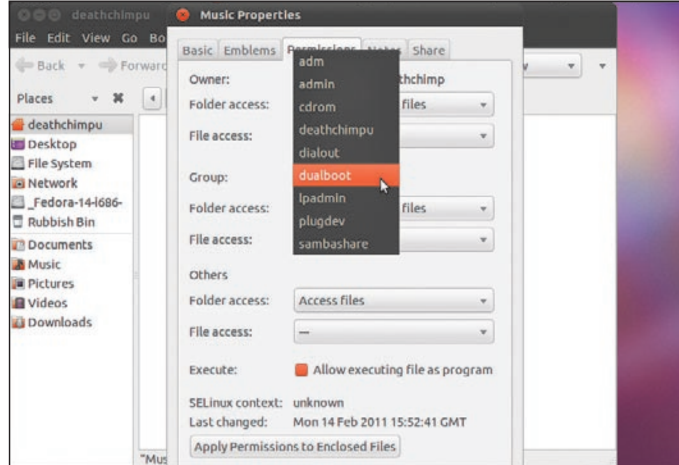
› Удалите старую папку Music, переименуйте Link To Music, и сможете слушать одну и ту же музыку в двух разных дистрибутивах.

Шаг за шагом: Делимся файлами в группе



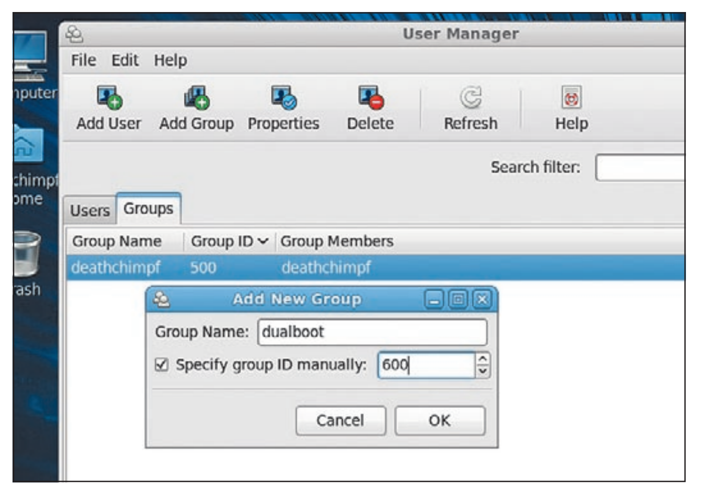
1 Создадим общую группу

Создайте новую группу и добавьте папки, которые вы хотите сделать общими для этой группы. Это намного удобнее, чем каждый раз для совместного использования заходить от имени root.



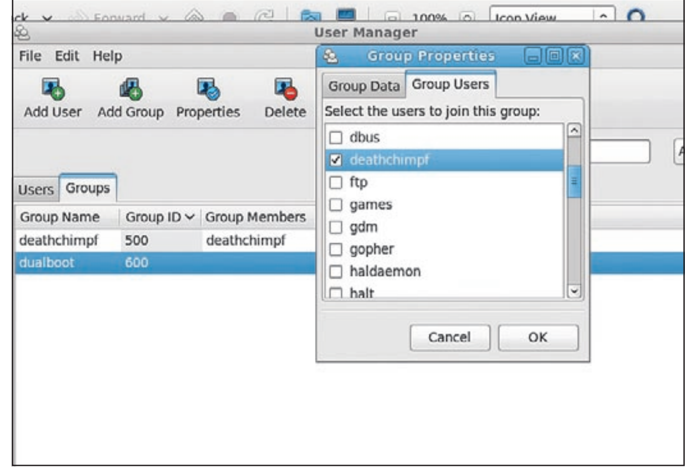
2 Сменим разрешения папок

Сейчас просто щелкните правой кнопкой мыши по папкам, которые нужно сделать общими, чтобы поменять их разрешения – сделать доступными для группы, созданной на предыдущем шаге.



3 Добавим группу в Fedora

Перезагрузитесь в Fedora и создайте здесь точно такую же группу. Самое важное, чтобы GID'ы в обоих дистрибутивах были одинаковыми.



4 Добавим пользователя Fedora в группу

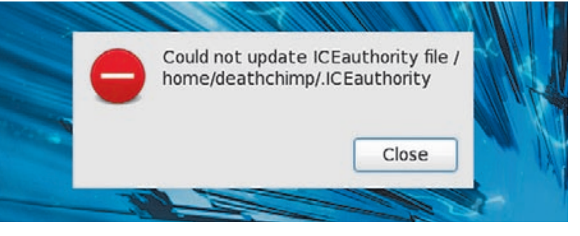
Не забудьте добавить пользователя Fedora в созданную вами группу, иначе вы не сможете добавлять в нее файлы.

Перезагрузитесь в Fedora и зайдите в System > Administration > Users and Groups. Щелкните по вкладке Groups, а затем Add Group. Как и в Ubuntu, следует создать группу 'dualboot' с групповым ID 600, но на сей раз перейти в Properties > Group Users и добавить в группу пользователя Fedora (у нас это deathchimpf). Выйдите из системы и снова зайдите, после чего смените права доступа у **Music** так же, как это делалось в Ubuntu.

В установке Ubuntu откройте терминал и введите:

```
gksudo nautilus
```

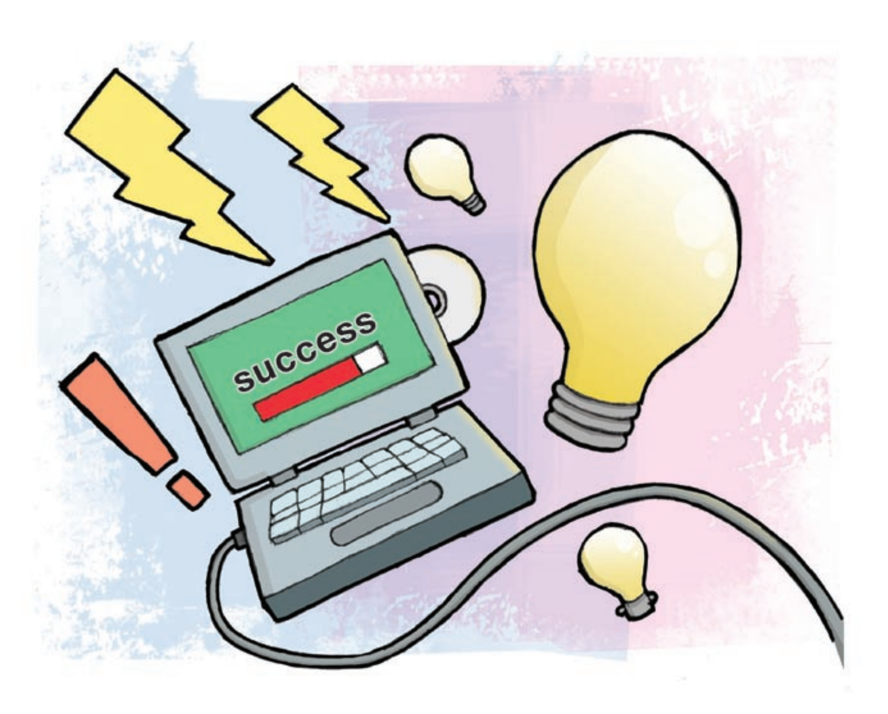
Введите ваш пароль, и файловый менеджер *Nautilus* откроется с правами root, что позволит вам работать с папками, принадлежащими обоим созданным пользователям (в нашем случае это deathchimpf и deathchimpu – так проще запоминать). Перейдите к deathchimpf, нажмите Ctrl+N, чтобы открылось новое окно *Nautilus*, и в новом окне перейдите в deathchimpu. Сейчас перед вами должны лежать два домашних каталога. Щелкните правой кнопкой по **Music** в deathchimpf и выберите Make Link. Полученная папка должна получить название **Link To Music** и иметь в своем значке маленькую стрелочку, которая указывает, что это ссылка, а не настоящая папка. Все, что остается сделать – перетащить ссылку к deathchimpu, удалить у него старую папку **Music** и переименовать **Link To Music** в **Music**. А сейчас – вперед: поделите и другие файлы между дистрибутивами! LXF



› Вот что бывает, если не создать разные учетные записи пользователя для каждой установки: непонятное сообщение об ошибке и неисправная система.

Todoyu: Рулим

Шашанк Шарма устанавливает *Todoyu*, мощный многопользовательский и многообещающий инструмент управления проектами.





Наш эксперт

Шашанк Шарма
Свыше четырех лет пишет о свободном ПО для различных изданий, включая Linux.com. Он также является соавтором книги *Beginning Fedora*.

Складывайте слой за слоем один на другой, и в итоге получится нечто креативное. Или, на худой конец, нечто высокое. И вообще, если малость порассуждать абстрактно, почти все в жизни – будь то небоскреб или сэндвич – результат нагромождения слоев.

А если слоями являются задачи, и много различных задач вместе составляют проект, то менеджер проекта обязан обеспечить, чтобы все слои укладывались в должный срок. Для этого ему надо следить за развитием каждой из частей процесса.

Как и водится в сообществе свободного кода, понаделано много удивительных приложений, удовлетворяющих данную потребность. Одна из таких программ – *Todoyu*, многопользовательский инструмент управления проектами, который помогает следить за статусом вашего молодого продукта, от планирования и до завершения. *Todoyu* умеет отслеживать отдельные задачи, обмениваться файлами, поддерживать календари и план отпусков, учитывать расходы по какой-либо конкретной деятельности и многое другое – и все это в удобном окне браузера.

Оговорки и обещания

Вариации фразы «расход бензина у каждого свой» нередко встречаются в описаниях инструментов управления проектами: мы применяем их по различным причинам, а стало быть, и требования у нас рознятся. Но если вы еще не нашли свой идеал решения, *Todoyu*, возможно, станет ответом на ваши молитвы.

Впрочем, сперва предупредим: *Todoyu* не слишком прост для освоения, чему также не способствует нехватка приличной документации. Главная печаль, однако, его зависимость от PHP 2.2.x, способная застопорить попытки ввести *Todoyu* в рабочий процесс.

Если у вас сравнительно свежая версия любого дистрибутива, весьма вероятно, что в его репозитории предлагается только ветвь PHP 2.3.x. Конечно, Deb- и RPM-пакеты для 2.2.x найти легко; но 2.2.x имеет еще и собственные зависимости, например, *PHP-common*, и гоняться за ними за всеми – занятие довольно утомительное.

Кроме того, при установке PHP следует включить функциональность *mbstring* – это можно сделать через репозиторий программного обеспечения вашего дистрибутива. В Fedora потребуется следующая команда:

```
su -c "yum install php-mbstring"
```

В Ubuntu *mbstring* – не независимый пакет, а часть *php5-dev*; поэтому вы должны выполнить такую команду:

```
sudo apt-get install php5-dev
```

Помимо PHP, понадобятся также *Apache* и *MySQL*, но уж их-то установить из репозитория легко.

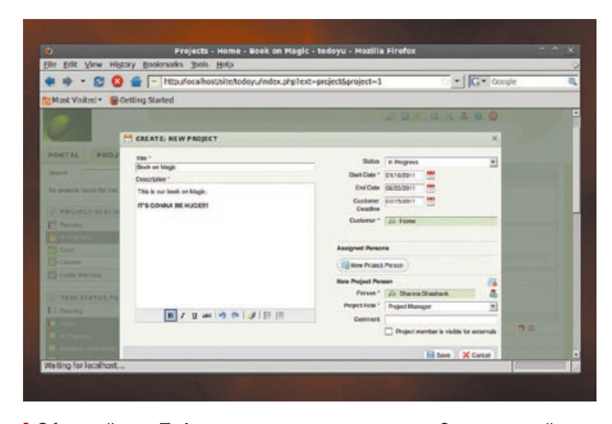
Теперь скачайте последний архив с сайта и распакуйте его. В зависимости от вашего дистрибутива и архиватора по умолчанию, файл **.htaccess** вам может потребоваться извлечь вручную, поскольку не все архиваторы умеют извлекать его из архива, и это может вызвать некоторые проблемы с установкой и настройкой *Todoyu*.

Приступим

Итак, вы установили *Todoyu*, и теперь, понятное дело, желаете увидеть предлагаемые красоты. Чтобы начать работу, просто наберите в браузере http://yourserver/todoyu/index.php и войдите в систему.

Если во время установки вы выбрали импорт демо-данных, ваша версия *Todoyu* будет забита примерно так же, как популярное заведение быстрого питания в обеденный перерыв. Хотя эти данные отчасти заменяют документацию, лучше работать с ними на локальной установке *Todoyu*, чтобы не засорять рабочую область.

Процесс создания собственного проекта в *Todoyu* не похож на многие аналогичные инструменты, где первым шагом являет-



› Обустройство *Todoyu* отнимет немало времени. Зато, с нашей помощью и проявленным упорством, вы получите мощный менеджер проектов.

проектами

ся создание нового проекта и заполнение его шаг за шагом. Мало того, что имеется ряд обязательных полей – таких как название, даты начала и окончания и так называемый Customer [Заказчик]: вам придется продираться через обеспечение *Todoyu* массой других основных данных, включая информацию о роли, деятельности и прочем. Но не робейте: на самом деле ввод достаточно прост. Займемся этой задачей.

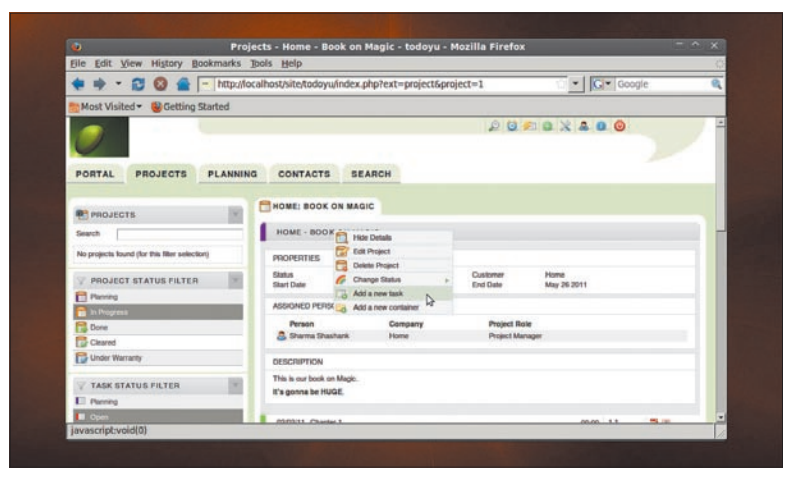
Войдя в интерфейс *Todoyu*, вы увидите кучу иконок в правом верхнем углу экрана. Обратите внимание на две из них, практически у центра: одна называется Admin [Администратор] и помечена скрещенными гаечным ключом и отверткой, а другая – Create [Создать], в виде зеленого кружка с белым крестом посередке. Ими мы и воспользуемся. Суть других отобразят подсказки при наведении на них мышью.

Добавляем роли

В *Todoyu* роли используются для описания различных категорий работников. Скажем, если ваш проект – издание книги, сопутствующими ролями будут писатели и редакторы. По сути, каждый род работы требует отдельной роли. Затем вы назначаете исполнителей по каждой категории.

Чтобы добавить новую роль, щелкните по значку Admin на панели инструментов в верху экрана, создайте иконку и выберите новую роль. Затем добавляйте исполнителей на эту роль, нажав на кнопку Role Members [Исполнители] в нижней части. Ролей мы пока не создали, поэтому просто заполните заголовок и описание. Установите флажок Active, затем нажмите кнопку Save [Сохранить] в нижнем правом углу. Создайте все роли, требуемые проектом, и пометьте их все как активные.

До сих пор мы говорили только о глобальных ролях, но надо создать роли и по отношению к самому проекту. Например, нужен,



› **Каждая запись по проекту предоставляет контекстное меню, облегчающее присвоение заданий, редактирование деталей и изменение общего статуса.**

как минимум, руководитель проекта. На странице Admin щелкните на Все Записи в левой панели, затем нажмите кнопку в крайнем правом поле проекта ролей. Нажмите Create New Project Roles [Создать новые роли проекта] и введите название новой роли.

Теперь нужно ввести Activity Type [Вид деятельности] – это зависит от типа проекта. В примере с книгой мероприятия будут такими, как ее написание, редактирование и создание оригинал-макета. Чтобы создать новый вид деятельности, перейдите на страницу Admin, нажмите All Records [Все Записи] слева, а затем нажмите кнопку в дальнем углу поля Activity Types. Теперь нажмите кнопку Create New Activity Type [Создать новый Вид деятельности] и повторяйте эти шаги до создания полного перечня. »



Кнопка Создать вверху окна не предложит опцию Новая роль, пока вы не перейдете на страницу Администратор.

Интерфейс Todoyu

Основной интерфейс *Todoyu* представлен пятью вкладками: Portal, Projects, Planning, Contacts и Search. Хотя некоторые из них говорят сами за себя, огромное количество функций, предлагаемых каждой, означает, что стоит остановиться на них более подробно.

› Portal [Портал]
Портал – это приборная доска *Todoyu*. Здесь вы получаете доступ к проектам, задачам, спискам и другой связанной с проектом информации.

› Projects [Проекты]
В *Todoyu* можно работать сразу с несколькими проектами. На левой панели выбирается, какие задачи посмотреть по какому-либо конкретному проекту. На главном экране вверху показаны данные о проекте, а в нижней части – все относящиеся к нему задачи. Задачи отмечены цветами, что позволяет легко отличать готовые от выполняемых.

› Planning [Планирование]
Согласно названию, здесь планируются события на указанный день, неделю или месяц. Перейдите на вкладку, и в левой панели отобразится текущий календарь, а главный экран предоставит планировщик на день. Между днем, неделей или месяцем переключайтесь вкладками в верхней части экрана. Чтобы добавить событие, дважды щелкните на соответствующнм дне или времени и заполните необходимую информацию. В выпадающем списке Event [Событие] перечислены практически все типы событий, включая праздники и дни рождения.

› Contacts [Контакты]
Здесь вы найдете список всех контактов, добавленных в *Todoyu*; люди и организации разнесены по отдельным вкладкам. Отметим, что хотя во вкладке Контакты можно просмотреть подробную информацию о любых людях или компаниях и даже обновить свою информацию, новые записи здесь создавать нельзя. Кроме того, в поле поиска слева поддерживается частичное соответствие, и вам не нужно вводить полные данные. *Todoyu* сверяет введенный текст со всеми доступными полями, такими как имя и фамилия, адрес электронной почты и т.д., и отображает список всех совпадений.

› Search [Поиск]
Наконец, одной из самых удивительных особенностей *Todoyu* является вкладка Поиск – это очень сложный набор инструментов. Вы можете создавать фильтры на основе различных параметров задач или проекта в целом. Такие элементы, как Created by [Кем создано], Owned by [Принадлежит], Deadline [Срок выполнения] и Статус [Status] – лишь часть доступного. Найдя искомое, можно сохранить настройки поиска, нажав на кнопку Save As New Filter [Сохранить как новый фильтр] на левой панели.

Роли имеют права

Возможно, вы не захотите давать одинаковые права всем участникам проекта – например, не каждому сотруднику дозволено редактировать и удалять задачи, так зачем предоставлять им такую возможность? *Todoyu* требует, чтобы для человека, вовлеченного в проект, была указана, по крайней мере, одна роль; права определяются ролью, и не надо выдавать их людям индивидуально. Иными словами, назначив права на роль, вы наделите ими всех людей в данной роли.

Скорая помощь

Для неактивных ролей права определить нельзя. Поэтому, создавая роли, не забудьте нажать Active [Активна].

Чтобы назначить какой-либо роли права, нажмите кнопку Admin, а затем Rights & Roles [Права и роли] в левой панели. В закладке Rights [Права] вы найдете выведенными на экран все определенные вами роли. Мы будем работать с ролью Редактор, поэтому щелкните Редактор и выберите модуль для этой роли в выпадающем списке Extension [Расширение]. Вы найдете там модули для календарей, персональных настроек, задач проекта и управления, и многое другое.

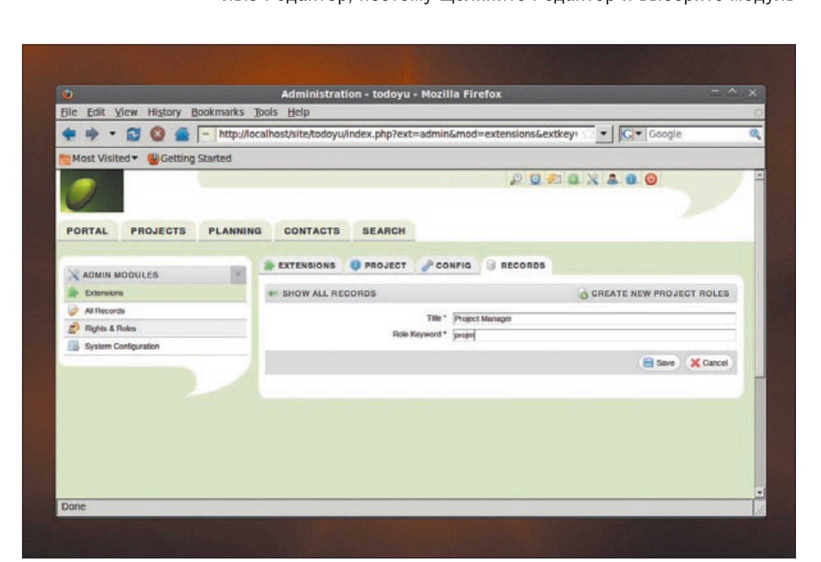
> «Можно даже назначить одного человека на несколько ролей.»

В зависимости от выбранного модуля, внизу страницы появится целая куча настроек и флажков для определения разрешений. Например, Sysmanager: модуль System Manager позволяет определять разрешения для редактирования роли и расширений. Если вы хотите выдать нескольким ролям одинаковые права, пометьте их, используя клавишу Ctrl, выберите модуль из выпадающего списка и определите их права одним щелчком.

После этого подготовительная работа будет почти закончена. Пора добавлять [Лица] – фактических исполнителей по проекту. Для этого нажмите кнопку Create [Создать] на панели инструментов в верхней части страницы и выберите New Person [Новое лицо].

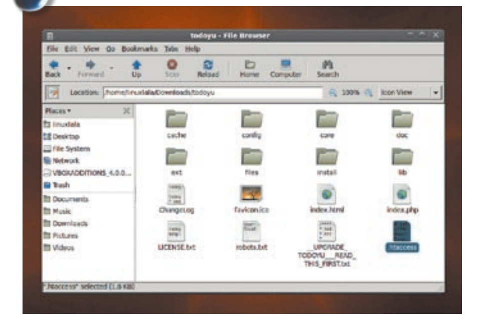
Диалоговое окно Create: New Person [Создать: Новое лицо] сложностей не вызывает. Просто заполните всю информацию по порядку. Можно внести и рабочие, и личные данные для каждого работника – в *Todoyu* это просто. Например, чтобы добавить адрес для нового контакта, нажмите кнопку New Address [Новый Адрес]. Выберите Home [Дом] или Office [Офис] в раскрывающемся списке Address Type [Тип адреса] и заполните все поля, такие как почтовый индекс и город. Чтобы добавить другой адрес для этого работника, просто повторите процесс.

Введя личные данные сотрудника, нажмите кнопку Enable Login [Разрешить Вход], если хотите дать ему доступ к *Todoyu*: тогда он сможет следить за статусом проекта. Добавьте имя пользователя, пароль и адрес электронной почты, и нажмите кнопку New Role [Новая роль] внизу. Выберите роль для этого лица – и ваш первый работник настроен и готов приняться за выполнение своих обязанностей. Процесс можно повторить для создания нескольких записей, в зависимости от кадрового состава проекта. Можно даже назначить одного человека на несколько ролей. Просто нажмите на кнопку Новая роль, выберите другую роль из выпадающего списка – и готово.



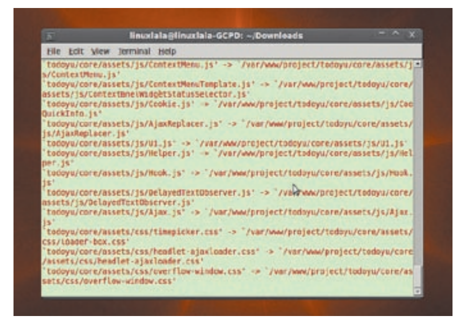
› **Требуются не только роли по работе над проектом, но и несколько ролей, напрямую связанных с самим процессом.**

Шаг за шагом: Установка Todoyu



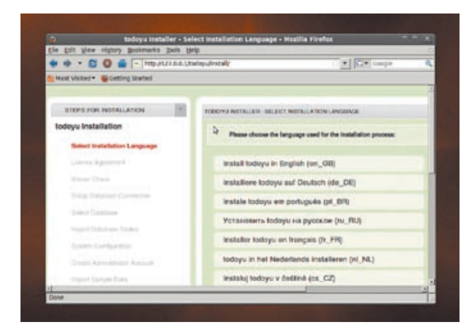
1 Скачайте архив

Последний релиз *Todoyu* – 2.0.7. Скачайте архив и распакуйте его. Убедитесь, что ваш архиватор добыл файл **.htaccess** из архива, поскольку на форумах поддержки описано множество случаев, когда отсутствие этого файла вызывало проблемы у пользователей. В процессе извлечения создастся каталог **Todoyu-2.0.7**.



2 Загрузите каталог

Переименуйте каталог Todoyu-2.0.7 в **todoyu** и загрузите его в корневой каталог документов вашего сервера. Теперь нужно разрешить запись в нескольких файлах и каталогах, поэтому сделайте **chmod** для следующих: **/config**, **/cache**, **/files** и **index.html**. Этот этап очень важен: если его сейчас опустить, возникнут проблемы с установкой.



3 Установите в браузере

Для начала установки введите в браузере http://yourserver/todoyu. Процесс выполняется шаг за шагом и включает выбор языка, принятие лицензии (для *Todoyu* это BSD), и так далее. Убедитесь, что вы правильно ввели пароль и имя пользователя базы данных *MySQL*. Кроме того, будьте осторожны с данными учетной записи администратора в *Todoyu*.

» **Пропустили номер?** Узнайте на с. 107, как получить его прямо сейчас.

В зависимости от вашего проекта, вам, возможно, придется нанимать работников из разных фирм. Например, если вы строите себе дом, вам нужны архитектор, водопроводчики и электрики, а также строители стен и крыши. Вероятно, это люди из разных учреждений.

Таким образом, кроме добавления этих людей в качестве сотрудников, можно также рассмотреть вопрос о создании для них пункта Company [Компания]. Это не обязательный шаг, но он многое упрощает. Вся эта информация уже должна у вас быть, остается лишь ввести ее в *Todoyu*.

Для создания Company, нажмите кнопку Create и выберите новую компанию. Заполните информацию, затем нажмите кнопку New Person рядом с сотрудниками. Поле Person [Лицо] поддерживает автоматическое заполнение и подскажет вам имена, когда вы начнете ввод. Если вы еще не создали нужного человека, нажмите кнопку Add New Person [Добавить новое лицо] в правой части поля Person.

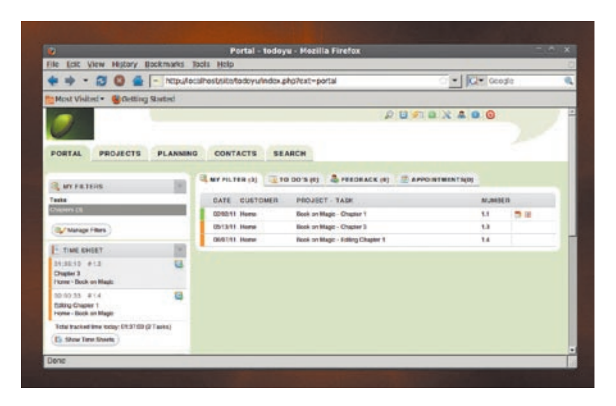
Ваш первый проект

Введя в *Todoyu* всю необходимую для нашего проекта основную информацию, приступим к его созданию. Нажмите кнопку Create, выберите New Project [Новый проект] и задайте подходящее название с описанием. В правой части диалогового окна Create: New Project введите другие детали, такие как Customer и даты начала и окончания проекта.

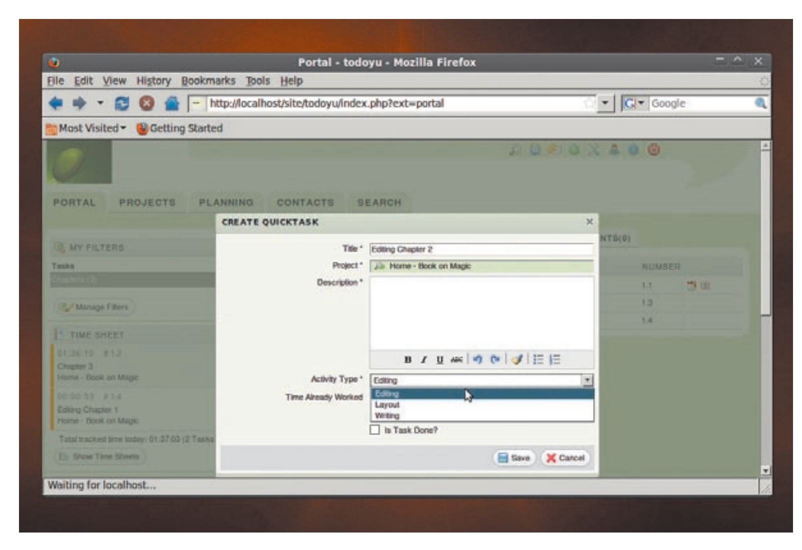
Обратите внимание, что поле Customer также принадлежит к автозаполняемым, и если вы не ввели данные для лица или компании, заказавших данный проект, на предыдущих шагах, вы не сможете закончить настройку. По сути, автозаполнение поля может создать определенное препятствие, потому что не позволяет просто ввести какое попало имя как «затычку»: все, что вы выбираете, уже должно быть частью основных данных, введенных ранее.

Заполнив информацию о заказчике, выберите Assigned Persons [Назначенные лица] > New Project Person. Наконец, выберите роль этого человека в поле Project Role [Роль проекта] и нажмите кнопку Save внизу для создания проекта. Это вернет вас в основной интерфейс *Todoyu*. Перейдите на вкладку проекта слева, чтобы просмотреть вновь созданный проект.

Теперь можно добавлять задачи. Для создания новой задачи можно нажать кнопку Create вверху и выбрать новую задачу, или щелкнуть правой кнопкой мыши на названии проекта и нажать кнопку Add A New Task [Добавить новую задачу]. Заполните информацию и выберите статус из раскрывающегося списка – к ним относятся Done [Выполненные], Rejected [Отклоненные] и In Progress [Текущие]. Наконец, укажите примерное время на выполнение проекта.



› Слежение за временем — просто одна из функций *Todoyu*. Можно запустить и остановить его в заголовке табелей учета рабочего времени на левой панели, во вкладке Портал либо Проекты.



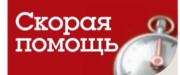
› Используйте кнопку Quicktask вверху для создания новой задачи, независимо от экрана, где вы находитесь.

Следите за задачами

Вы можете отслеживать время, затраченное на каждую задачу, при помощи функции слежения за временем в *Todoyu*, которая работает только с текущими задачами. Перейдите на вкладку Projects в основном интерфейсе *Todoyu*. В панели слева нажмите на ссылку Progress [Прогресс] под заголовком Task Status Filter [Фильтр статуса задач], который выведет все текущие задачи. Вы можете нажать на имя задачи для просмотра более подробной информации.

В нижней части нажмите кнопку Start [Пуск] на вкладке Time Tracking [Слежение за временем]. Для данной задачи запустится таймер. Это особенно полезно, когда одно и то же лицо назначается на несколько задач, так как позволяет легко отслеживать время, отведенное на каждую из них. При перемещении на другую задачу, просто остановите таймер для задач в работе.

К каждой задаче также можно добавить комментарии или прикреплять соответствующие файлы. Чтобы прикрепить к задаче файл, нажмите на имя задачи и на вкладку Add First Asset [Добавить первый актив] в нижней части экрана. Нажмите кнопку Browse [Обзор] и выберите файл, который хотите загрузить. LXF



Руководство по быстрому старту можно скачать на панели слева от страницы входа http://yourserver/todoyu/index.php. Это не полный учебник, но, тем не менее, полезный.

Слишком много данных

Todoyu можно настроить для работы сразу над несколькими проектами. С виду это полезно, но при таком подходе отсутствует сегрегация введенных данных. А именно, если вы добавите 20 человек в один проект и пять в другой, *Todoyu* будет всегда показывать все 25 человек. То же относится и к компаниям.

В малых проектах это терпимо, но в больших уже будет означать лавину информации. К счастью, расширенные опции поиска помогут придать этой лавине смысл – без них вы бы пропали.

Чтобы лучше понять, что имеется в виду, создайте тестовую копию *Todoyu* и выберите во время установки импорт демо-данных. Создастся тестовая копия *Todoyu*, заполненная тремя проектами и всякими другими данными. Осмотритесь, и вы скоро поймете, что это неподъемно для обработки.

Потому-то мы и рекомендовали для рабочей установки *Todoyu* игнорировать опцию импорта демо-данных: информация из примеров будет мешать вашей работе.



Майк Сондерс ведет ТРЕНИНГИ LINUX

Часть 3: От аппаратуры и загрузки переходим к каталогам файловой системы, разделам и разделяемым библиотекам. И – да: в конце вас ждет тест!

Mike



Linux Professional Institute

Наш эксперт

Майк Сондерс пишет о Linux более десяти лет, и установил больше дистрибутивов, чем съел горячих обедов.

Отгадайте, сколько файлов окажется на жестком диске после установки Debian 6 с одного компакт-диска? Подумайте – мы подождем. А ответ такой: 82698. Верится с трудом, и таким количеством файлов невозможно управлять, но, к счастью, в каталогах файловой системы Linux эта масса файлов располагается весьма комфортно, и у каждого файла есть осмысленное место для жизни. Какова задача каждого отдельного файла, слишком подробно знать не обязательно, но расположение файла определяет его общее назначение в глобальном устройстве системы.

На данном уроке мы рассмотрим, как устроена файловая система Linux, как разбить жесткий диск на разделы и изменить конфигурацию загрузчика *Grub*. Мы также увидим, как разделяемые библиотеки повышают безопасность и экономят место на жестком диске. Как и в других статьях этой серии, некоторые каталоги и команды могут различаться в зависимости от используемого дистрибутива. Однако для обучения мы рекомендуем дистрибутив, соответствующий промышленному стандарту, такой как Debian, которым мы и пользовались при написании этой статьи. Установить его можно с **LXFDVD**.

Раздел 1: Файловая система Linux

Для начала разберемся, как установка Linux располагается на жестком диске. В противоположность Windows, где есть различные «стартовые точки», или буквы диска, для каждого устройства, в Linux для всего есть только один источник – нечто вроде «Большого Взрыва» данных. Это **/**, или корневой каталог (**root**), который не следует путать с пользователем root, администратором системы. Все остальные каталоги, такие как **/home/username**, ваш домашний каталог, ответвляются от него. Другими словами, корневой каталог – это каталог верхнего уровня, и все остальные каталоги в системе являются его подкаталогами. Вот какие каталоги вы найдете в корневом:

» **/bin** Содержит двоичные файлы, представляющие собой исполняемые программы. Это важнейшие системные утилиты, такие как **ls**, **df**, **rm** и т. д. Здесь должны находиться все файлы, необходимые для загрузки системы и исправления неполадок, в то время как каталог **/usr/bin** имеет другое назначение – какое, мы скоро увидим.

» **/boot** Здесь находится файл образа ядра (**vmlinuz** – z означает, что файл сжат); это программа, загружаемая и исполняемая загрузчиком. Здесь также находится образ RAM-диска (**initrd**), который предоставляет ядру минимальную файловую систему и набор драйверов для запуска системы. Во многих дистрибутивах здесь же сидит файл **config** с настройками компиляции ядра, а также подкаталог **grub** с конфигурацией загрузчика.

» **/dev** Файлы устройств. К большинству устройств в Linux можно обращаться так, как если бы они были файлами, считывая и записывая в них байты. Загляните в первую часть этой серии, чтобы узнать подробнее о **/dev**.

» **/etc** Преимущественно файлы настройки в текстовом формате, хотя есть и исключения. Загрузочные скрипты (см. вторую часть этой серии) также находятся здесь. Здесь находятся системные конфигурационные файлы для таких программ как *Apache*; конфигурация настольных программ обычно находится в каталоге **home**.

» **Месяц назад** Мы пролили свет на аппаратуру и процесс загрузки.

```
root@debian:/boot/grub#
root@debian:/boot/grub# ldd /usr/bin/gedit
        linux-gate.so.1 =>  (0xb77b9000)
        libgtksourceview-2.0.so.0 => /usr/lib/libgtksourceview-2.0.so.0 (0xb774c
000)
        libgtk-x11-2.0.so.0 => /usr/lib/libgtk-x11-2.0.so.0 (0xb7376000)
        libgdk-x11-2.0.so.0 => /usr/lib/libgdk-x11-2.0.so.0 (0xb72df000)
        libatk-1.0.so.0 => /usr/lib/libatk-1.0.so.0 (0xb72c3000)
        libgdk_pixbuf-2.0.so.0 => /usr/lib/libgdk_pixbuf-2.0.so.0 (0xb72aa000)
        libcairo.so.2 => /usr/lib/libcairo.so.2 (0xb7235000)
        libgio-2.0.so.0 => /usr/lib/libgio-2.0.so.0 (0xb7199000)
        libpango-1.0.so.0 => /usr/lib/libpango-1.0.so.0 (0xb7154000)
        libgobject-2.0.so.0 => /usr/lib/libgobject-2.0.so.0 (0xb7116000)
        libgmodule-2.0.so.0 => /usr/lib/libgmodule-2.0.so.0 (0xb7112000)
        libgthread-2.0.so.0 => /usr/lib/libgthread-2.0.so.0 (0xb710d000)
        libgconf-2.so.4 => /usr/lib/libgconf-2.so.4 (0xb70da000)
        libglib-2.0.so.0 => /lib/libglib-2.0.so.0 (0xb7010000)
        libX11.so.6 => /usr/lib/libX11.so.6 (0xb6ef3000)
        libSM.so.6 => /usr/lib/libSM.so.6 (0xb6eeb000)
        libICE.so.6 => /usr/lib/libICE.so.6 (0xb6ed4000)
        libm.so.6 => /lib/i686/cmov/libm.so.6 (0xb6eae000)
        libpthread.so.0 => /lib/i686/cmov/libpthread.so.0 (0xb6e95000)
        libc.so.6 => /lib/i686/cmov/libc.so.6 (0xb6d4e000)
        libpangocairo-1.0.so.0 => /usr/lib/libpangocairo-1.0.so.0 (0xb6d43000)
```

› Как показывает запуск команды ldd, большинство программ наследуют значительную часть своего функционала от разделяемых библиотек.

» **/home** Здесь обычно находятся домашние каталоги (**home**). У каждого пользователя здесь есть каталог с персональными файлами и настройками.

» **initrd.img** Символическая ссылка (это не настоящий файл – больше похоже на ярлык в Windows) на вышеупомянутый RAM-диск в **/boot**. Полный путь, на который она ссылается, можно посмотреть командой **ls -l**.

» **/lib** Разделяемые библиотеки; подробнее о них см. во врезке «*Что такое разделяемые библиотеки?*» внизу страницы. Как и в каталоге **/bin,** здесь находятся важнейшие библиотеки, необходимые для загрузки и запуска системы на базовом уровне. В каталоге **/lib** также находятся модули ядра (см. первую часть этой серии уроков).

» **/lost+found** Если во время крупной операции записи данных на диск падает система или отключается электричество, и при следующем запуске система выполняет проверку диска (**fsck**), фрагменты частично утерянных файлов помещаются сюда.

» **/media** При подключении внешнего диска, скажем, USB-брелка, происходит автоматическое монтирование, и в этом каталоге появляется подкаталог с содержимым этого диска.

» **/mnt** Немного похож на **/media**, за исключением того, что обычно используется для смонтированных вручную долговременных хранилищ данных, включая жесткие и сетевые диски.

» **/opt** Дополнительное ПО. Этот каталог используется очень редко, но в некоторых дистрибутивах и пакетах вы найдете здесь большие пакеты вроде KDE и *OpenOffice.org*. Это позволяет хранить все в одном месте и, следовательно, более удобно удалять или обновлять ПО помимо менеджера пакетов.

» **/proc** Доступ к информации о процессах. Здесь можно увидеть каждый процесс (выполняющуюся задачу) в системе, что и поддерживает философию Unix «всё является файлом».

» **/root** Персональные файлы суперпользователя, он же – root. Многие администраторы хранят здесь резервные копии файлов настройки. В многопользовательских системах жизненно важно, чтобы у обычных пользователей не было сюда доступа.

» **/sbin** Как и в каталоге **/bin**, здесь находятся двоичные исполняемые файлы, но доступ к ним разрешен только суперпользователю. Это программы, которые не должны запускать обычные пользователи: утилиты настройки сети, форматирования разделов и т. д.

» **/selinux** Место для файлов, используемых средой Security-Enhanced Linux.

» **/srv** Этот каталог предназначен для данных, обслуживаемых системой (например, web-сервером). Однако большинство программ вместо него пользуются **/var**.

» **/sys** Нечто вроде осовремененного варианта **/dev** с дополнительными возможностями. Здесь можно получить информацию об устройствах и ядре, но для обычных задач администратора этот каталог не так важен.

» **/tmp** Временные файлы. В этот каталог может записывать данные любая программа, и вы увидите здесь случайные файлы фоновых сервисов, web-браузеров и т. д. В большинстве дистрибутивов он очищается при загрузке.

» **/usr** Это другой мир. **/usr** содержит собственные версии каталогов **bin**, **sbin** и **lib**, но они предназначены для приложений, существующих вне базовой системы. Все жизненно важное для запуска системы должно находиться в **/bin**, **/sbin** и **/lib**, а необязательные программы вроде *Firefox* и *Emacs* должны жить здесь. И на то есть веская причина: разместив базовую систему на одном разделе (**/**), а дополнительные программы на другом (**/usr**), систему можно сделать более гибкой. **/usr**, например, можно монтировать через сеть. Здесь также помещен каталог **/usr/local**, где обычно находятся программы, которые вы компилируете сами, чтобы их не видел менеджер пакетов.

» **/var** Здесь находятся файлы, которые часто меняются. Другими словами, это файлы, в которые часто записываются данные – системные журналы, базы данных и файлы почтовых клиентов. В большинстве дистрибутивов здесь также размещается корень документа Apache (**/var/www**). На мощных серверах, которые пишут данные интенсивно, он часто выносится в особый раздел с настройками файловой системы, повышающими скорость.

» **vmlinuz** Символическая ссылка на файл образа ядра в **/boot**.

В зависимости от дистрибутива вы можете найти в каталоге **root** и другие файлы. Однако большинство дистрибутивов стараются следовать положениям стандарта Filesystem Hierarchy Standard (FHS). Это попытка унифицировать каталоги файловой системы между различными дистрибутивами. Вы сможете погрузиться глубже в **/var**, **/usr** и другие каталоги, заглянув на справочную страницу иерархии каталогов в терминале командой **man hier**. »

Что такое разделяемые библиотеки?

Библиотека – это фрагмент кода, который не является исполняемым, но может использоваться другими программами. Например, пусть вы создаете программу, которая должна будет разбирать XML-файл, но не хотите писать XML-парсер. Вместо этого вы можете воспользоваться *libxml*, библиотекой для обработки XML, которую уже написали другие. В обычной установке Linux есть сотни таких библиотек, включающих библиотеки для базовых функций C (*libc*) и графические интерфейсы (*libgtk*, *libqt*).

Библиотеки можно связать с программой статически, «зашив» в окончательную версию исполняемого файла, но обычно это разделяемые файлы в каталогах **/lib**, **/usr/lib** и **/usr/local/lib** с **.so** в имени файла, что означает «разделяемый объект» [shared object]. Благодаря этому несколько программ могут одновременно пользоваться одной и той же библиотекой, и если в ней обнаружится уязвимость, достаточно исправить ее только один раз. С разделяемыми библиотеками уменьшаются размеры исполняемых файлов, экономя дисковое пространство. Командой **ldd** можно узнать, какие библиотеки используются программой. Например, **ldd /usr/bin/gedit** выводит список библиотек, включающих:

```
libgtk-x11-2.0.so.0 => /usr/lib/libgtk-x11-2.0.so.0
(0xb7476000)
```

Gedit зависит от *GTK*, поэтому ему нужна *libgtk-x11*, и справа вы видите местоположение этой библиотеки в системе. Но что определяет расположение библиотек? Ответ – файл **/etc/ld.so.conf**, который ссылается на файлы в **/etc/ld.so.conf.d**. Эти файлы содержат текстовые строки с расположением библиотек в файловой системе, такие как **/usr/local/lib**. Если вы устанавливали библиотеки куда-либо, можете добавить новые файлы с их расположениями, но после этого запустите **ldconfig** (от имени root) чтобы обновить кэш, используемый загрузчиком программ.

Иногда бывает нужно запустить программу, использующую библиотеку с нестандартным расположением. Для этого воспользуйтесь переменной **LD_LIBRARY_PATH**. Например, следующая команда запустит исполняемый файл **myprog** из текущего каталога и на время добавит **mylibs** к списку стандартных путей к библиотекам:

```
LD_LIBRARY_PATH=/path/to/mylibs ./myprog.
```

Так во многих играх связываются библиотеки отдельно от исполняемого файла без его наличия.

Раздел 2: Схемы разбиения диска на разделы

Разбиение диска на разделы – одна из тех задач, за которые приходится браться нечасто, но которые имеют далеко идущие последствия. Если неправильно выделить место для какого-то раздела, неприятности вам обеспечены. Особенности разбиения диска на разделы зависят от установщика вашего дистрибутива, и мы не будем перечислять здесь все возможные варианты. Вместо этого воспользуемся утилитой разбиения диска на разделы, которая есть во всех дистрибутивах, и рассмотрим некоторые схемы разбиения диска.

Откройте терминал, переключитесь на пользователя root и введите команду:

```
fdisk /dev/sda
```

Замените **sda** устройством, соответствующим вашему диску. (На компьютерах с одним жестким диском это **sda**, а если Linux загружается со второго диска – **sdb**. Для верности загляните в вывод **dmesg**.) **fdisk** — простая командная утилита разбиения диска. Как и *Vi*, выглядит она аскетично, но умеет все. Введите **p**, и вы увидите список разделов диска как на экранном снимке ниже. Введя **m**, вы получите список доступных команд: разделы можно удалять (**d**), создавать новые (**n**), сохранять изменения на диск (**w**) и т. д.

Это мощная утилита, но она предполагает в вас знание того, что вы делаете, и опекать вас не будет. **fdisk** также не форматирует разделы – им просто дается номер типа. Для форматирования раздела наберите **mkfs** и нажмите Tab, чтобы просмотреть варианты завершения команды. Вы увидите, что есть команды для форматирования разделов в традиционных Linux-форматах (таких как **mkfs.ext4**), в формате Windows FAT32 (**mkfs.vfat**) и других.

Наряду с разделами файловой системы существует еще раздел подкачки (**swap**). Он используется для организации виртуальной памяти. Другими словами, когда программа больше не помещается в микросхемах оперативной памяти, потому что память съели другие программы, ядро может вытеснить эту программу в раздел подкачки, записав содержимое памяти в виде данных на этот раздел. Когда программа снова становится активной, она возвращается с диска назад в оперативную память. Волшебной формулы точного расчета размера раздела подкачки не существует, но администраторы советуют устанавливать его в размере вдвое больше объема оперативной памяти, но не свыше 2 ГБ. Отформатировать раздел подкачки можно командой **mkswap** с устройством в качестве параметра (например, **/dev/sda5**), а активировать – командой **swapon** с устройством в качестве параметра. В качестве раздела подкачки можно также использовать один файл – более подробную информацию ищите на man-страницах команд **mkswap** и **swapon**.

```
mike@debian: /usr/share/man/man7
File  Edit  View  Terminal  Help
root@debian:~# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): p

Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0006a080

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1        3792    30450688   83  Linux
/dev/sda2            3792        3917     1003521    5  Extended
/dev/sda5            3792        3917     1003520   82  Linux swap / Solaris

Command (m for help):
```

› **У большинства дистрибутивов свои графические утилиты для разбиения диска, но верная *fdisk* найдется везде.**

Скорая помощь

Не хотите, чтобы другие пользователи залезали в настройки *Grub* во время загрузки? Защитите их паролем, и только вы сможете их менять – как это сделать, можно узнать из онлайн-документации *Grub* по ссылке http://tinyurl.com/6czhkn8.

Схемы разбиения диска на разделы

Итак, как разбить диск на разделы? Есть три общих подхода:

1 Все в одном. Наличествует один большой раздел с файлами операционной системы, домашними каталогами, временными файлами, данными серверов и всеми остальными файлами. Иногда это не самый эффективный подход, но он самый простой, и у всех каталогов равные возможности по использованию дискового пространства. Во многих настольных дистрибутивах этот подход используется по умолчанию.

2 Разделяем корневой раздел и **home**. Чуть более сложная схема, в которой каталог **/home** размещается в собственном разделе, отдельно от корневого раздела (**/**). Большое преимущество этого подхода в том, что можно обновлять, переустанавливать и менять дистрибутивы, при необходимости полностью уничтожая все файлы операционной системы при сохранении персональных данных и настроек в **/home** нетронутыми. Более подробно о преимуществах обособления каталога **/home** смотрите в учебнике про Grub в этом журнале.

3 Разделы для всего. При работе на критически важном компьютере, вроде открытого извне web-сервера с круглосуточным режимом, вы можете применить несколько весьма эффективных схем разбиения. Например, пусть у вашего компьютера два жестких диска: медленный и быстрый. Если вы работаете на бойком почтовом сервере, можете поместить корневой каталог на медленный диск: ведь он используется только при загрузке системы. А **/var/spool** лучше поместить на быстрый диск, так как каждую минуту с ним производятся сотни операций чтения и записи.

Эта гибкость в разбиении диска на разделы – большое преимущество Linux и Unix, и польза от этого все более и более возрастает. Возьмем, например, самые современные быстрые диски SSD: каталог **/home** можно разместить на обычном жестком диске, обеспечив себе большой объем пространства по низкой цене, а каталог **root** — на SSD, и система будет загружаться и выполнять программы со скоростью света.

Проверьте себя!

При продвижении по этой серии уроков вы, возможно, хотите попутно проверять свои знания. В конце концов, если после нашего расставания вы направитесь на полноценную сертификацию LPI, вы должны уметь пользоваться знаниями на лету, не заглядывая в руководства. В конце серии мы подготовим для вас полный список примерных вопросов, а пока вот несколько задач и вопросов по материалам пройденных нами трех разделов:

1 Где находятся образ ядра и файлы RAM-диска?

2 Объясните отличия в назначении каталогов **/lib**, **/usr/lib** и **/usr/local/lib**.

3 Расскажите о возможных схемах разбиения диска на разделы в Linux. Почему стоит размещать каталог **/home** на отдельном разделе?

4 Опишите процесс добавления новых каталогов с библиотеками в систему.

Попробуйте ответить на эти вопросы, не обращаясь за подсказками к соответствующим разделам. Если вы сомневаетесь, не волнуйтесь – просто вернитесь и прочтите их снова. Лучший способ усвоить полученные знания – взять их на вооружение и поупражняться на собственном компьютере (или на каком-нибудь дистрибутиве в *VirtualBox*, чтобы не подвергать опасности свою систему).

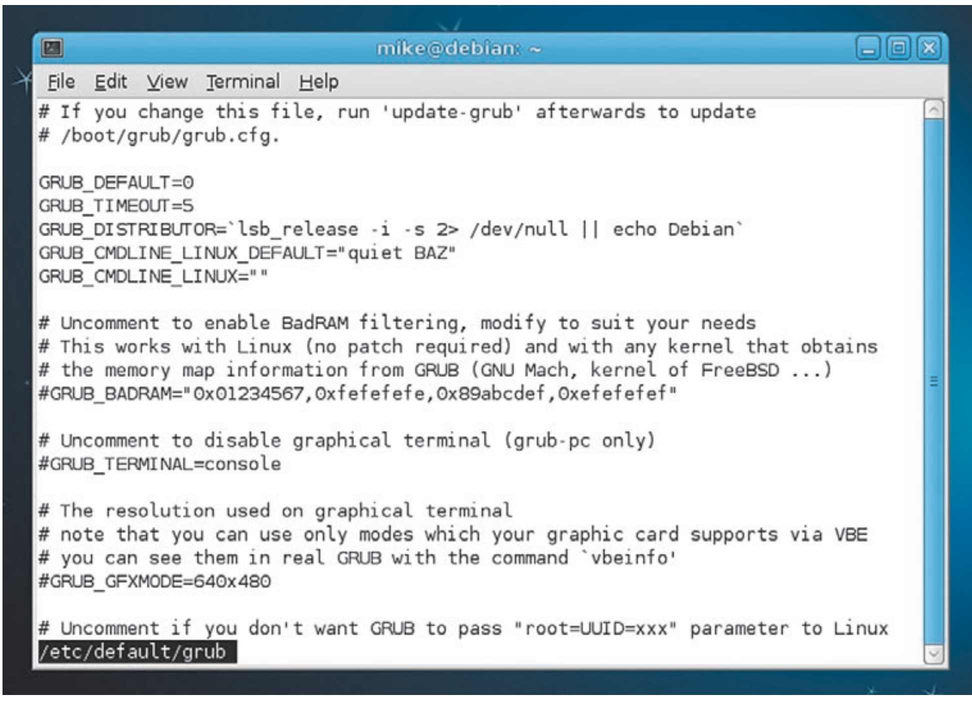
» **Пропустили номер?** Узнайте на с. 107, как получить его прямо сейчас.

Раздел 3: Настройка загрузчика

Для загрузки ядра и старта всего процесса загрузки системы почти во всех дистрибутивах ныне используется *Grub 2* (полностью – *Grand Unified Bootloader*). В предыдущей статье мы познакомились с самим *Grub* и с тем, как изменить из него параметры загрузки. Но эти изменения временные. Если что-то нужно менять каждый раз, то прерывание каждой загрузки может обернуться настоящей головной болью. Решение – изменить файл **/etc/default/grub**.

Вообще-то это не файл настройки *Grub* – тот называется **/boot/grub/grub.cfg** и автоматически создается скриптами после обновления ядра, поэтому изменять его вручную не следует. В большинстве случаев требуется добавить тот или иной параметр к строке загрузки ядра, чтобы отключить определенное устройство или загрузиться в определенном режиме. Сделать это можно, открыв файл **/etc/default/grub** в текстовом редакторе от имени пользователя root и взглянув на следующую строку:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
```



› Добавлять настройки ядра нужно в файл /etc/default/grub — не забудьте потом запустить update-grub, чтобы настройки записались в /boot/grub/grub.cfg.

Она содержит параметры по умолчанию, которые передаются ядру Linux. Добавьте в двойные кавычки после **quiet** необходимые опции, разделяя их пробелами. Сделав это, выполните команду:

```
/usr/sbin/update-grub
```

Она обновит **/boot/grub/grub.cfg**, добавив в него новые параметры.

Стар и груб

Если у вас более старый дистрибутив с *Grub 1*, настройка будет несколько иной. У вас будет файл **/boot/grub/menu.lst** с записями наподобие

```
title Fedora Core (2.6.20-1.2952.fc6)
root (hd0,0)
kernel /vmlinuz-2.6.20-1.2952.fc6 ro root=/dev/md2 rhgb
quiet
initrd /initrd-2.6.20-1.2952.fc6.img
```

Здесь можно добавить опции прямо в конец строки **kernel**, сохранить файл и перезагрузить компьютер, чтобы изменения вступили в силу. Если *Grub* испортится или его удалит другой загрузчик, вы сможете переустановить его, выполнив следующую команду от имени root:

```
grub-install /dev/sda
```

Замените **sda** на **sdb**, если хотите установить его на второй жесткий диск. Эта команда записывает начальную часть *Grub* в первые 512 байт жесткого диска, также известных как главная загрузочная запись (MBR). Обратите внимание, что *Grub* не обязательно устанавливать в MBR; его можно установить в суперблок (первый сектор) раздела, позволяя основному загрузчику в MBR подгрузить дочерний загрузчик. Впрочем, скорее всего, эти сведения выходят за рамки LPI 101, и вы вряд ли с ними столкнетесь; но знать об этом стоит.

Наконец, хотя *Grub* используется в подавляющем большинстве дистрибутивов, все же есть и такие, в которых еще засиделся более старый загрузчик LILO – Linux Loader [Загрузчик Linux]. Его файл настройки – **/etc/lilo.conf**, и после любого изменения файла нужно запустить **/sbin/lilo**, чтобы обновить настройки в загрузочном секторе. LXF



Точка монтирования – место соединения раздела с файловой системой. Предположим, например, что раздел **/dev/sdb2** будет использоваться для домашних каталогов. В этом случае его точка монтирования – **/home**. Просто!

Волшебство /etc/fstab

Познакомившись со схемами разбиения диска на разделы и узнав, как создавать разделы с помощью команд **fdisk** и **mkfs**, вы, возможно, заинтересовались, как это устроено в рабочей системе Linux. Во главе данной схемы стоит файл **/etc/fstab**, текстовый файл, где описана связь разделов с точками монтирования. Загляните в этот файл, и вы увидите несколько строк наподобие такой:

```
UUID=cb300f2c-6baf-4d3e-85d2-9c965f6327a0 / ext3
errors=remount-ro 0 1
```

Строка состоит из пяти полей. Первое поле – устройство, задаваемое в виде **/dev/sda1-type** или в более современном формате идентификатора UUID (это уникальный идентификатор устройства; его можно добыть командой **blkid**). Затем указана точка монтирования – в данном случае это корневой каталог. За ней следуют тип файловой системы и опции монтирования.

Приведенные здесь параметры означают, что если при монтировании диска загрузочными скриптами возникают ошибки, его нужно перемонтировать в режиме только для чтения, чтобы операции записи не причинили дальнейшего ущерба. Списки доступных опций для каждого типа файловой системы имеются на man-странице команды **mount**. Последние числовые параметры связаны с проверками файловой системы, и изменять их значения по умолчанию не нужно.

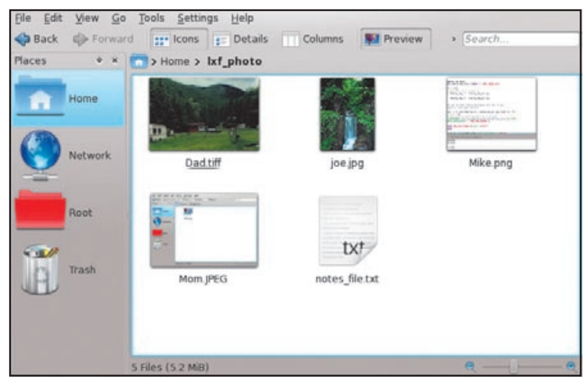
Вы можете добавить свои точки монтирования в **/etc/fstab** в текстовом редакторе, но учтите, что в некоторых дистрибутивах файл автоматически перезаписывается, поэтому обязательно сохраните исходную копию.

» Через месяц Управление пакетами и RPM, и Deb!

ExifTool: Фото по

Беспорядок в фотографиях может причинить вам немало беспокойства, отчего и захворать недолго. К счастью, **Марко Фиоретти** нашел лекарство.





› Когда вы просите лжедрузей и родных пополнить вашу коллекцию фотографий, получается, естественно, бардак.

Наш эксперт

Марко Фиоретти Автор книги "*The Family Guide to Digital Freedom*", а также активист свободного ПО и программист.

Не так давно фотографии представляли собой карточки из глянцевой бумаги с цветными пятнами на них, которые с годами бережно накапливались в картонных коробках и красивых альбомах. Сегодня появление доступных цифровых фотокамер привело к тому, что каждый год мы можем делать сотни или тысячи новых снимков. Несмотря на то, что они цифровые, их тоже нужно хранить.

Утилиты, подобные *Digikam*, несмотря на свои возможности, не самый лучший вариант для организации хранилища. Поэтому в этом руководстве я решил поделиться с вами полуавтоматической технологией, которая позволит переименовать и отсортировать фотографии и поместить их в архив. В ее основе – утилита *ExifTool* и два простых скрипта. Все это вы найдете на нашем диске. Для работы с ними необходимо только общее представление, что такое скрипты и как они работают.

Какова роль скриптов?

Мои скрипты делают очень немногое, но если делать это вручную, уйдет масса вашего времени. Во-первых, они переименовывают файлы, помечая каждый из них датой. Затем они автоматически сохраняют каждую фотографию в каталоги, которые также иерархически разбиты по годам, месяцам и дням. С их помощью также можно добавить комментарии, рейтинги и метки к вашим фотографиям. Эти задачи часто лучше решают графические программы вроде *Digikam*, но если сначала упорядочить вашу коллекцию, то это сэкономит вам массу времени.

А кроме того, ими довольно просто пользоваться, если вы знаете, что у них внутри, и имеете некоторое представление о метаданных фотографий. По правде говоря, самое сложное в данной статье – это не понять, как работают скрипты, а уяснить, почему они должны делать то или иное.

Начинаем с хаоса

Итак, однажды вы проснулись и решили создать одну, тщательно упорядоченную цифровую коллекцию всех цифровых и бумажных фотографий, которые вам не безразличны. Делать это вручную – дело не просто долгое, но и безнадежное. Вероятно, вы столкнетесь со следующими задачами, в любом сочетании: первая – оцифровка бумажных фотографий, что скучно и требует времени, но несложно. Чтобы сделать все быстрее, можно сканировать по четыре фотографии за один раз с любым сканером формата A4 и сохранять результаты в файлах с уникальными именами – способ, описанным мной в **LXF123**.

Другая очевидная необходимость – разобраться во всех фотографиях, которые прислали друзья. Они могут быть сделаны разными камерами или даже такой же камерой, как у вас. В первом случае на диске появятся файлы со всеми известными человечеству и поэтому несовместимыми схемами именования. Сотни файлов будут иметь имена вроде **XYZ_001.JPG**, ..., **XYZ_153.JPG**, а другие – имена вроде **P00001.jpeg, ..., P99999.jpeg**; и т.д. Если вам не повезет и у друзей найдется аналогичная камера, есть риск получить несколько файлов с одинаковыми именами, перезапись которых обернется катастрофой, если не проявить должного внимания.

Наихудший сценарий появляется в случае объединения самых досадных характеристик двух вышеупомянутых проблем. В этом случае вы получаете файлы отсканированных фотографий во всех графических форматах, открытых со времен изобретения транзистора. Более того, в их именах не будет и следа пользы или смысла. Наверное, кто бы ни прислал вам эти файлы, он подсознательно вас ненавидит.

Поименовать всю эту неразбериху – наша первая задача. В конце концов, унифицированные имена файлов – основное условие для эффективной и портируемой системы управления

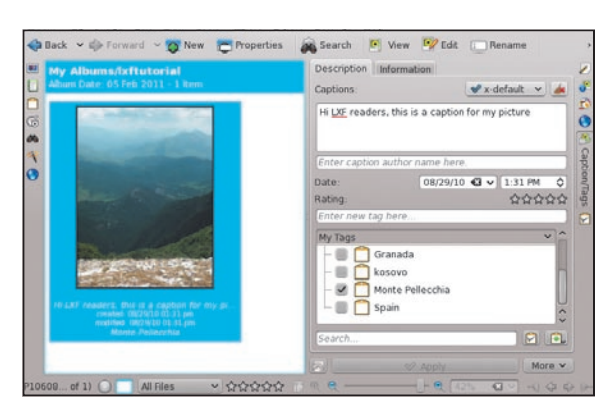
полочкам

фотографиями. Для этого есть несколько критериев, и главный из них в том, что фотографии должны остаться в отдельных файлах на диске в стандартных форматах. Если зарыть фотографии в глубинах базы данных, производительность улучшится, но этот вариант ни в коем случае не подходит, если вы хотите иметь возможность просто делать резервные копии, а также дальше укреплять коллекцию.

По похожим причинам, а также потому что имена файлов должны быть уникальны, избегайте в именах файлов символов, не являющихся буквами и цифрами. Описательные названия тоже не подходят – названия типа **Мамин_выпускной.jpg** могут иметь смысл сегодня, но с годами вы будете мучительно вспоминать: «От чьей это мамы? Моей или моей второй половинки?»

А как насчет возрастающей нумерации? О, нет. Она перестанет возрастать, а стало быть, иметь смысл, когда придется добавить фотографии в уже существующий альбом. Нет, наиболее переносимая, масштабируемая и защищенная на будущее схема именования файлов – давать каждому файлу имя в соответствии с точным временем его создания. Файлы с именами вида YYYYMMDDHHMMSS переносимы; имена не слишком длинные; их всегда можно отсортировать по дате в любой программе, даже если добавить или удалить фотографии из альбома; и из них очень удобно делать резервные копии, отсортированные в хронологическом порядке. Если поместить все фотографии, снятые в определенный день или в определенный месяц в один каталог с таким именем, как **Year/Month** или **YearMonth**, это только упростит индексирование.

Теперь, определившись со схемой именования файлов, поговорим о метаданных, то есть о дополнительной информации по каждой фотографии. В них может храниться почти все что угодно: от автора и копирайта до установок камеры или меток категорий. Существует три основных стандарта метаданных фотографий, но мы сосредоточимся на наиболее распространенном варианте – фотографии в формате JPEG с метаданными EXIF (Exchangeable Image File Format).



› В *Digikam* есть мощный интерфейс для создания тэгов и работы с метаданными, но работа мышью — не единственный способ.

Как быть с пробелами?

Основной скрипт, с которым мы имеем дело, не сработает, если в именах файлов попадутся пробелы или спецсимволы. Я умышленно исключил этот вариант, потому что, отказавшись от него, любой хороший и сложный скрипт можно написать гораздо быстрее и проще, хотя и Linux, и оболочка способны разобраться с такими символами. Поэтому я никогда не храню файлы со странными именами сам и переименовываю все файлы, полученные от других, при первой же возможности. Впрочем, вам не обязательно впадать в эти крайности. Чтобы обрабатывать файлы со странными именами, просто замените строку **for PIC...** в моем скрипте первыми четырьмя строками скрипта, который вы найдете по адресу http://www.cyberciti.biz/tips/handling-filenames-with-spaces-in-bash.html.

Метаданные фотографии могут храниться либо в файле (как это происходит с EXIF), либо в базе данных. Первый подход гарантирует, что они не будут утеряны при переносе фотографий. Второй значительно ускоряет поиск и другие операции. Лично я пользуюсь обоими подходами сразу: сначала записываю метаданные в формате EXIF в файлы, затем, открывая свои альбомы в *Digikam*, копирую эту информацию во внутреннюю базу данных.

Используем скрипты

Теперь перейдем к конкретному примеру – к ситуации, показанной на экранном снимке на предыдущей странице. Здесь есть несколько фотографий в различных форматах, которые нужно упорядочить, разобрать и опознать. Обратите внимание, что для ясности примера в нашем каталоге содержится только четыре файла фотографий (и один текстовый файл, назначение которого мы проясним чуть позже), но скрипт будет работать и с тысячами файлов. Некоторые из изображений – даже не фотографии, а просто случайные экранные снимки. Я сделал это намеренно, чтобы проиллюстрировать работу скриптов. При необходимости любую картинку можно преобразовать в нечто гораздо более полезное – файл, с которым будет работать любой менеджер фотографий (с метаданными и со всем остальным), как если бы этот файл был снят цифровой камерой.

Вот как работает скрипт: для начала он помещает все файлы, которые нужно отсортировать, в специально созданный для этого каталог. Затем – и это единственное, что вам придется сделать вручную – надо подготовить простой текстовый файл **notes_file.txt**.

Он необходим потому, что мы переименовываем, сортируем и архивируем фотографии на основе времени их создания. Для цифровых фотографий эти данные находятся в тэге EXIF под названием DateTimeOriginal. Во всех остальных случаях мы должны предоставить примерное значение. Эти значения мы и записываем в файл **notes_file.txt**:

```
$ cat notes_file.txt
joe.jpg | 2001:01:01
Mom.JPEG | 1998:12:01
Mike.png
Dad.tiff | 2005:03:28
```

Как видите, нужно только написать имена всех файлов фотографий, которые находятся в каталоге, по одному в каждой строке. Добыть их список можно командой **ls -l | cut -c45- > notes_file.txt**. Затем добавьте после символа | время создания каждой фотографии в формате YYYY:MM:DD. Если в соседних строках есть два или более файлов с одинаковыми датами, добавьте дату только к первому из них. Когда все будет готово, запустите в командной строке скрипт, передав ему полный путь к этому каталогу в качестве первого и единственного аргумента:

```
$ photo_archiver.sh $HOME/lxf_demo
```

Работу скрипта с каждым файлом отражает алгоритм, приведенный на рисунке справа.

Разбираем скрипт

Первое, что мы делаем – сохраняем исходную версию файла, делая рабочую копию. Затем, при необходимости, эта копия преобразуется в формат JPEG, так как нам нужен формат, позволяющий хранить тэги EXIF. Для этого мы пользуемся утилитой **convert**, входящей в состав пакета *ImageMagick* (www.imagemagick.org):

```
convert /tmp/my_tmp_photo /tmp/my_tmp_pic.jpg
```

Затем мы обрабатываем тэги EXIF с помощью *ExifTool*:

```
1 DATE=`exiftool -s -f -DateTimeOriginal /tmp/my_tmp_pic.jpg | cut -d: -f2-`
2
3 if [ "$DATE" == " -" ]
4 DATE_FROM_NOTES=`$RETRIEVE_DATE notes_file.txt $PIC`
5 if [ "$DATE_FROM_NOTES" == "NO_DATE_FOUND" ]
6 then
7 echo "NO_DATE_FOUND for $PIC, sorry!"
8 exit
9 else
10 exiftool -DateTimeOriginal="$DATE_FROM_NOTES 12:00:00" /tmp/my_tmp_pic.jpg
11 fi
12 fi
```

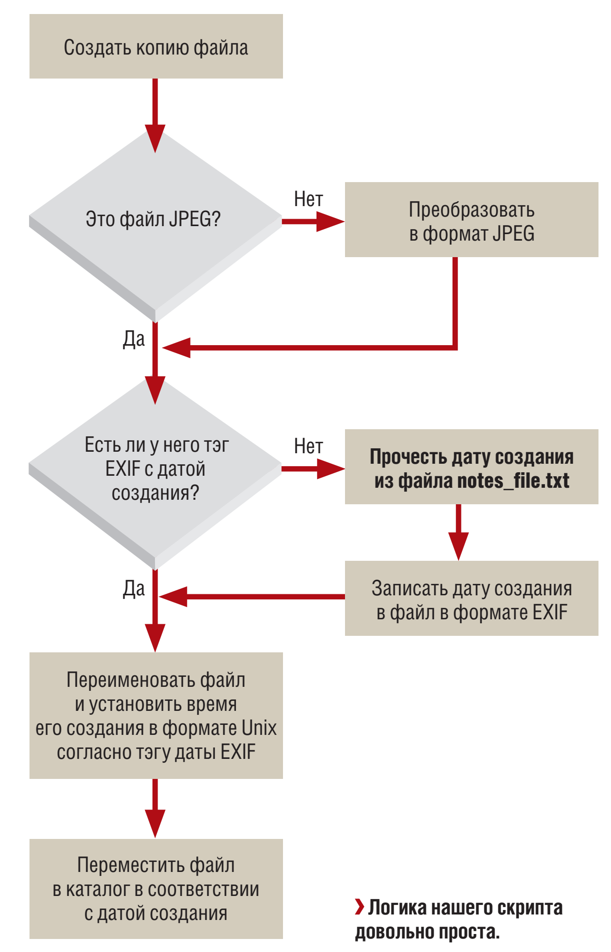
В первой строке *ExifTool* получает параметр EXIF под названием **DateTimeOriginal** из файла JPEG и сохраняет его, удалив все пробелы и ненужные фрагменты, в переменной **$DATE**. Если в результате этой операции мы получаем пустую дату, мы просим **retrieve_date_from_file.pl** (он есть на нашем DVD) извлечь подходящую дату из файла **notes_file. txt**. Затем с помощью *ExifTool* мы записываем это значение в файл JPEG. Обратите внимание, что скрипт всегда устанавливает одно и то же время (**12:00:00**). Если вам нужна большая точность, введите более точную дату в **notes_file.txt** и измените второй вызов *ExifTool* соответствующим образом.

Теперь у нас появился файл в формате JPEG **my_tmp_pic.jpg**, который содержит дату съемки в переменной EXIF **DateTimeOriginal**. Затем выполняется следующая часть скрипта:

Как удалить метаданные EXIF?

Запись комментариев, тэгов и другой информации в фотографии, говоря техническим языком, весьма здравый подход. Однако с точки зрения взаимоотношений он может стать губительным. Причина в том, что EXIF – всемирный открытый стандарт, и данные EXIF не шифруются. А значит, каждый, кто получит копию вашей фотографии, сможет прочесть и все ее метаданные. Запечатлеть таким образом те или иные факты из жизни – это прекрасно, и ничто вам не мешает обозначить комментарием «Лошара» портрет вашего дядюшки, но прежде чем отправлять ему фотографию по электронной почте или загружать ее на Flickr, сделайте копию и выполните следующую команду, которая очистит содержимое всех полей EXIF:

```
exiftool -all= uncle_portrait.jpg
```



› **Логика нашего скрипта довольно проста.**

```
1 NEWDATE=`exiftool -s -f -DateTimeOriginal /tmp/my_tmp_pic.jpg | cut -d: -f2- | tr -d " :"`
2 PARTIAL_DATE=`echo $NEWDATE | cut -c1-12`
3 SECNDS=`echo "$NEWDATE" | cut -c13-14`
4 mv /tmp/my_tmp_pic.jpg /tmp/$NEWDATE.jpg
5 touch -t "$PARTIAL_DATE.$SECNDS" /tmp/$NEWDATE.jpg
```

Сначала мы опять считываем дату, ликвидируем все пробелы и двоеточия (команда **tr -d** в конце первой строки) и называем результат **$NEWDATE**. Формат этой переменной – YYYYMMDDHHMMSS. Первые 12 символов этой строки, определяющие дату и время с точностью до минуты, записываются в **$PARTIAL_DATE**. Два последних символа, секунды, записываются в переменную **$SECNDS**. Скоро вы поймете, зачем это нужно.

Причина, по которой мы создали переменную **$NEWDATE**, очевидна из четвертой строки приведенного фрагмента: эта переменная нужна, чтобы переименовать временный файл в **$NEWDATE.jpg**. В пятой строке происходит последняя часть не такой уж и черной магии – в ней в то же значение устанавливается время создания файла командой **touch**. К сожалению, эта утилита распознает секунды только после точки. Поэтому нам пришлось создавать отдельно **$PARTIAL_DATE** и **$SECNDS**. В итоге после выполнения скрипта **Mom.JPEG** превращается в **19981201120001.jpg**:

```
$ ls -l 19981201120001.jpg
-rw-rw-r-- 1 marco marco 61909 Dec 1 1998 19981201120001.jpg
$ exiftool -s -f -DateTimeOriginal 19981201120001.jpg | cut -d: -f2- | tr -d " :" 19981201120000
```

Понятно, что я имел в виду? Теперь у нас есть файл, чьи имя, время создания и дата съемки полностью совпадают с EXIF! Последнее, что делает скрипт **photo_archiver.sh** – помещает каждую

» **Пропустили номер?** Узнайте на с. 107, как получить его прямо сейчас.

фотографию в каталог с подходящим именем, создавая его, если такого еще нет:

```
1 YEAR=`echo $NEWDATE | cut -c1-4`
2 MONTH=`echo $NEWDATE | cut -c5-6`
3 DAY=`echo $NEWDATE | cut -c7-8`
4
5 if [ ! -d "$BASE_PHOTO_DIR/$YEAR/$MONTH/$DAY" ]
6 then
7 mkdir -p $BASE_PHOTO_DIR/$YEAR/$MONTH/$DAY
8 fi
9 mv /tmp/$NEWDATE.jpg
$BASE_PHOTO_DIR/$YEAR/$MONTH/$DAY
```

В первых трех строках **$NEWDATE** разбивается на год, месяц и день. В строках с 5 по 7 при необходимости создается каталог в соответствии с этой датой, и файл перемещается туда. В моем случае **19981201120001.jpg** попадает в каталог **$BASE_PHOTO_DIR/1998/12/01**.

А теперь на шаг отступим и восхитимся результатами работы скрипта, показанными на экранном снимке внизу. На левой панели мы видим, что скрипт создал правильную иерархию альбомов в каталоге **$BASE_PHOTO_DIR (lxf_demo)** согласно имеющимся данным EXIF и содержимому файла **notes_file.txt**. На центральной панели и в левом верхнем углу мы видим, что имена файлов и время их создания были изменены соответствующим образом. Наконец, на панели данных EXIF мы видим, что *Digikam* распознает и отображает исходную дату создания фотографии, которую скрипт прописал в EXIF.

Что еще нужно знать?

Внимательные читатели заметят в коде явную ошибку. А вдруг найдется несколько фотографий с одинаковыми значениями поля **DateTimeOriginal**? Не перезапишет ли каждая их них предыдущую, если все они получат одинаковые имена вида **YYYYMMDDHHMMSS.jpg**? Не отчаивайтесь, взгляните на экранный снимок под окончательными результатами процесса, и вы увидите, что файлов два, а не один. Дело в том, что скрипт проверяет, существует ли уже файл с таким именем, и если да, то делает грязный, но эффективный ход. Имя файла – целое число, и если файл **N.jpg** уже есть, скрипт сохраняет текущий файл под именем **N+1.jpg**.

Конечно, в нашем мире секунды все еще идут только от 0 до 59. Что делать, если найдется более 60 фотографий с одинаковым значением поля **DateTimeOriginal**? Все просто – у всех этих фотографий будет одинаковое корректное время создания в формате Unix (как у **19981201120001.jpg**), но только у первых 60 имя файла будет соответствовать времени создания. Если это реальная, а не теоретическая проблема, код нетрудно изменить, сохранив уникальность имен, например, за счет дополнительных суффиксов вроде **19981201120001.aa.jpg**, **19981201120001.ab.jpg**…

Наконец, JPEG – это формат с потерей качества. Если у вас есть фотографии высокого качества, сделать их JPEG-версии очень удобно, но нужно также сохранить все исходные файлы и связать их с JPEG-копиями. Тогда – и это упражнение остается вам, потому что здесь просто не осталось на него места – нужно переименовать исходные файлы по тому же шаблону YYYYMMDDHHMMSS, но сохранить исходные расширения.

Запутались?

Если вы запустили скрипт и получили не совсем то, что ожидали, не отчаивайтесь. Во-первых, *Digikam* – как и любой другой менеджер фотографий с базой данных – может синхронизировать изменения с тэгами EXIF не сразу. В *Digikam* выберите Images > Reread Metadata From Images [Изображения > Перечитать метаданные из изображений] в главном меню, и вы увидите свои изменения.

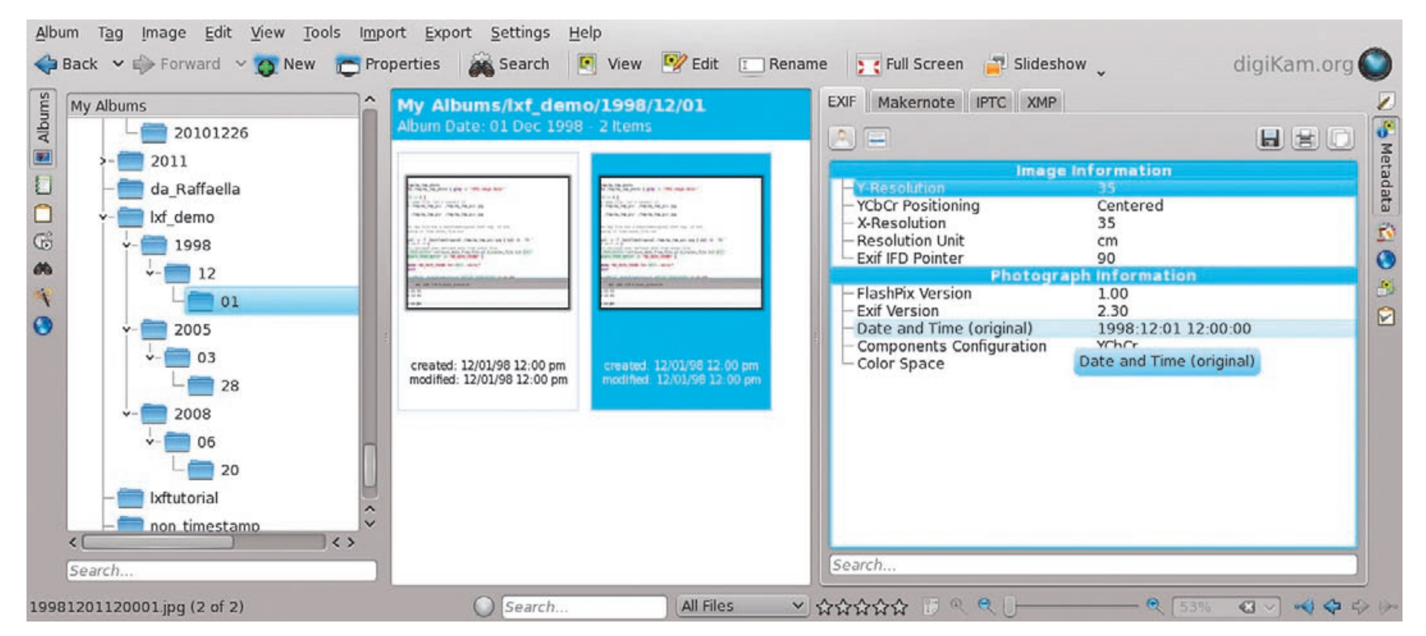
Также нужно знать, что и в EXIF, и в *Digikam* есть несколько тэгов с датой. Например, есть тэг, где хранится дата последнего изменения файла. Посмотрите на панель EXIF в *Digikam* или загляните в документацию по *ExifTool*, чтобы найти другие поля. Лично у меня никогда не было необходимости напрямую работать с дополнительными тэгами. Но если вы захотите это сделать, все, что нужно – добавить несколько строк в скрипт.

Как обобщить этот метод

Написать такой скрипт стало возможным благодаря единственному условию: способности *ExifTool* записывать и считывать тэги даты. Однако в метаданных EXIF можно хранить гораздо больше информации. Так как *ExifTool* поддерживает все параметры EXIF, с ее помощью можно автоматически устанавливать и любые другие параметры. Например, из командной строки можно добавить описание фотографии:

```
$ exiftool '-Description=this, instead, is a caption inserted with exiftool' P1090097.JPG
$ exiftool -Keywords='Montefeltro, family' P1090097.JPG
```

Возможности *ExifTool* на этом не заканчиваются. Я автоматически добавил к своим фотографиям широту и долготу с помощью скрипта из моего руководства по геотэггингу (в **LXF112**). Теперь в любой программе, поддерживающей работу с геотэгами, можно увидеть, где были сняты мои фотографии. Ну не отличная ли штука *ExifTool*? LXF

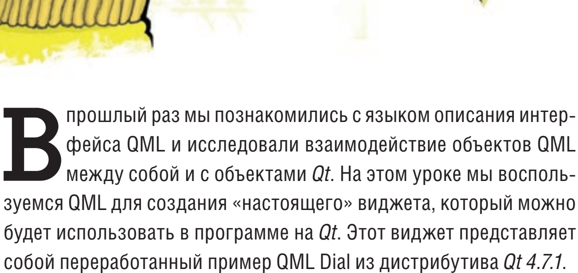


› **Окончательный результат: у всех файлов приличные имена, тэги и расположение. Красотища!**

QML: На службе

Андрей Боровский готов поспорить, что QML стал любимой игрушкой кодеров *Qt*, и все их усилия нацелены на рисование красивых виджетов QML.







› **Рис. 1. Исходный виджет в программе QML.**



Наш эксперт

Андрей Боровский
Человек, умеющий программировать на всех уровнях – от нестандартного «железа» до пользовательского интерфейса.

В прошлый раз мы познакомились с языком описания интерфейса QML и исследовали взаимодействие объектов QML между собой и с объектами *Qt*. На этом уроке мы воспользуемся QML для создания «настоящего» виджета, который можно будет использовать в программе на *Qt*. Этот виджет представляет собой переработанный пример QML Dial из дистрибутива *Qt 4.7.1*.

Сразу предупреждаю, что примеры из этой статьи будут работать с *Qt 4.7.1*, *4.7.2* и, надеюсь, с более поздними версиями *Qt*, но не с более ранними. Также хочу напомнить, что при работе с QML в файл проекта *Qt* нужно добавить строчку

```
QT += declarative
```

Демо QML Dial из дистрибутива *Qt* – это самостоятельная программа, написанная на QML (я уже писал о том, что QML можно использовать как самостоятельный язык программирования, наподобие JavaScript). Меня же, как программиста *Qt*, больше интересует применение QML для расширения возможностей интерфейсов программ *Qt*. Вот такое расширение мы сегодня и напишем, а заодно познакомимся с новыми возможностями *Qt* и некоторыми новыми методами взаимодействия *Qt* и QML. Сам объект QML Dial представляет собой почти фотореалистичную имитацию стрелочного индикатора, такого как спидометр или индикатор давления жидкости в трубе (рис. 1). Мы не только «перетащим» этот индикатор в свою программу *Qt*, но и дополним его некоторыми элементами (рис. 2).

> «Важнейшая задача QML – создание красивых интерфейсов.»

Как мы уже знаем, важнейшей задачей QML является создание красивых графических интерфейсов, а это, как вы понимаете, невозможно без мощных средств работы с растровой графикой (для программиста рисовать красивые интерфейсы внутрипрограммно, векторными функциями, несколько утомительно). Для работы с изображениями, хранимыми в файлах, в языке QML есть объект Image{}. Этот объект подобен тэгу HTML <img…>, за исключением того, что он может гораздо больше. В простейшем варианте использование объекта Image{} выглядит так:

```
Image { source: "background.png" }
```

Эта конструкция загружает фон нашего индикатора (описание внешнего вида индикатора хранится в файле **Dial.qml**, расположенном в директории **Dial** и сопутствующих файлах с расширением **png**.) На первый взгляд тут все понятно без особых пояснений. Мы загружаем изображение из файла **background.png**... Стоп. А куда мы его загружаем? Где оно будет расположено? Вспомним, что визуальная часть QML основана на Qt Graphics View Framework, а эта система по умолчанию стремится расположить графические элементы так, чтобы геометрический центр сцены совпадал с центром окна вывода. Так что по умолчанию изображение **background.png** просто займет место в центре окна нашего виджета (что нам и требуется).

Помимо собственных свойств, к каковым относится, например, свойство source, объект Image{} унаследовал от базовых объектов QML такие свойства как x, y, scale, rotation, transform, anchors и многие другие. Сам движок, выполняющий отрисовку изображения, обладает многими полезными возможностями.

приложений Qt

Например, если загружаемый формат поддерживает альфа-канал, то при выводе изображения учитывается уровень прозрачности его элементов.

Рассмотрим некоторые свойства объекта Image{}. Если мы хотим выложить сцену фоновым рисунком как плиткой, надо добавить свойство

```
fillMode: Image.Tile
```

Свойства scale, rotation и transform, как вы уже догадались, позволяют выполнять преобразования изображения, такие как масштабирование и вращение. Вот как, например, выполняется вращение тени под стрелкой нашего индикатора:

```
transform: Rotation {
        origin.x: 9; origin.y: 67
        angle: needleRotation.angle
    }
```

Вся эта конструкция находится в теле объекта image, отвечающего за вывод тени под стрелкой (у нас ведь почти фотореализм, так что тень должна двигаться вместе со стрелкой). В объекте Rotation мы задаем координаты центра вращения и угол поворота (если кто не понял, в этом чудесном языке двоеточие является оператором присваивания).

А как выполняется вращение самой стрелки? Тут все еще интереснее. Вот как выглядит описание стрелки:

```
Image {
      id: needle
      x: 98; y: 33
      smooth: true
      source: "needle.png"
      transform: Rotation {
        id: needleRotation
        origin.x: 5; origin.y: 65
        //! [needle angle]
            angle: root.angle
        Behavior on angle {
          SpringAnimation {
            spring: 1.4
            damping: .15
          }
        }
        //! [needle angle]
      }
  }
```

Большая часть этого описания должна быть вам уже понятна. Мы устанавливаем координаты стрелки; присвоение значения true свойству smooth приводит к тому, что при выполнении преобразований изображения стрелки (в нашем случае это вращение) выполняется специальная фильтрация, сглаживающая эффекты «лесенки», которые могут в ходе этих преобразований возникнуть. Сравните координаты стрелки и координаты тени. Между прочим, поскольку на изображении (файл **needle.png**) стрелка смотрит вверх, наш индикатор по умолчанию указывает на значение в середине диапазона. Можно было бы изменить это, повернув стрелку на картинке или выполнив необходимое вращение при инициализации, но мы оставим все как есть, потому что так интереснее.

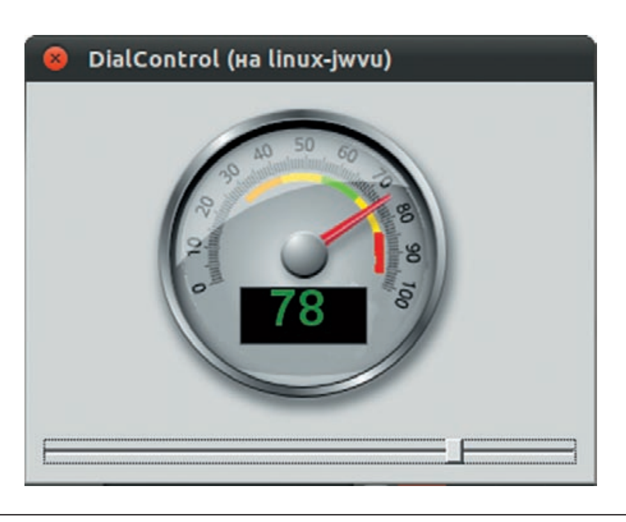
Описание объекта Rotation начинается так же, как и в случае с тенью, однако дальше следует странное. Конструкция Behavior on angle {} указывает, что должна делать стрелка, когда ее угол поворота меняется. Объект SpringAnimation создает специальный анимационный эффект при движении стрелки – эффект затухающих колебаний. Свойство spring указывает, насколько велика должна быть изначальная амплитуда колебаний, а свойство damping определяет скорость их затухания. В результате стрелка нашего индикатора будет колебаться как настоящая стрелка на пружине.

С помощью свойства visible, которым обладают все объекты QML, в том числе и объект image, мы можем управлять видимостью этих объектов.

Возможно, вас удивляет, что свойство angle объекта needleRotation меняется всякий раз, когда меняется свойство root.angle. В таких языках, как C++, одна операция присваивания означает одно изменение значения; но в QML, похоже, операция присваивания продолжает работать постоянно, и изменение значения свойства в правой части выражения присваивания приводит к изменению значения свойства в левой части, когда бы оно ни случилось. И это действительно так. В QML этот механизм назван связыванием свойств [property binding]. Строка

```
angle: root.angle
```

связывает между собой свойства root.angle и angle, так что изменение значения первого свойства всегда будет приводить к изменению значения второго. Связывание свойств – это особая форма присваивания, которая используется всегда, когда слева от оператора присваивания указано свойство объекта, а справа – любое синтаксически корректное выражение языка JavaScript (свойство, функция и т. д). То есть фактически связывание свойств является стандартным методом присваивания в QML, при котором свойство, стоящее слева от оператора присваивания, становится псевдонимом выражения, стоящего справа (вот почему связывать можно не только свойства, но и такие «пассивные» объекты, »



› **Рис. 2. Виджет на его основе в нашей программе *Qt*.**

как функции, которые сами не могут ничего инициировать). Связывание свойств – это тот механизм, который позволяет объектам QML обмениваться данными друг с другом, подобно тому, как объекты *Qt* обмениваются данными друг с другом с помощью сигналов и слотов. Точно так же, как в случае сигналов и слотов, связывание уже связанных свойств можно изменить с помощью элемента QML PropertyChanges {}. Мне очень нравится связывание свойств, и если бы я проектировал новый объектно-ориентированный язык программирования, то обязательно включил бы в него этот механизм.

Дальше в исходном примере на основание стрелки накладываются колпачок и стекло с бликом, изображения которых хранятся в файле **overlay.png**. Обратите внимание, что практически везде мы используем способность формата PNG создавать частично прозрачные изображения, иначе такой сложный элемент управления у нас просто не получился бы. Обратите внимание также на то, что порядок расположения изображений поверх друг друга соответствует порядку следования представляющих их объектов в тексте программы QML. В общем-то это естественно, если учесть, что виджет создается по мере выполнения программы.

> «Движок умеет работать с пространством ресурсов приложения.»

Ко всему этому великолепию я добавил совсем немного.

С объектом Rectangle мы уже встречались:

```
Rectangle {
x: 61
y: 118
width: 80
height: 36
color: "black"
border.color: "#888888"
 }
```

Новое здесь – свойство border.color, позволяющее задать цвет границы прямоугольника. Объект Text дублирует значение, которое указывает индикатор.

```
Text {
  color: "green"
 text: root.angle/2 + 50
  x: 80
  y: 114
  font.pointSize: 24; font.bold: true
  style: Text.Raised
  styleColor: "black"
}
```

Здесь я хотел бы обратить ваше внимание только на один момент. Поскольку по умолчанию стрелка индикатора смотрит вверх, фактические углы поворота следует указывать относительно этого положения, причем повернутая на некоторый угол стрелка при новом повороте интерпретирует новый угол так, как если бы она находилась в исходном положении.

Создаем виджет

Перейдем ко второй части нашей работы – превращению модуля QML в виджет *Qt*. Необходимые для этого основные операции мы изучили в прошлый раз. Но и теперь нам есть что добавить.

Представителем виджета QML в нашей программе *Qt* является класс Dial (файлы **Dial.h**, **Dial.cpp**). Здесь я приведу только объявление класса, остальное вы найдете на диске.

```
class Dial : public QObject
  {
   Q_OBJECT
   Q_PROPERTY(int angle READ angle WRITE setAngle NOTIFY
angleChanged)
 public:
   Dial();
   int angle();
   void setAngle(int a);
 signals:
   void angleChanged();
 private:
   int m_angle;
 };
```

Тем, кто читал предыдущую статью и знает *Qt*, тут все должно быть понятно. Класс экспортирует единственное свойство angle. Напомню только, что метод setAngle() должен явным образом эмитировать сигнал angleChanged(), иначе виджет QML никогда не узнает, что угол изменился. Знатоки сигналов и слотов *Qt* могут спросить, почему в сигнале angleChanged() не передается новое значение угла поворота. Если бы сигнал предназначался для других классов *Qt*, я бы так и сделал, но сигнал предназначен для виджета QML, а соответствующий объект этого виджета все равно прочитает значение свойства angle с помощью метода angle() (указанного после ключевого слова READ в макросе Q_PROPERTY). Так что передавать какой-либо параметр в сигнале *Qt* просто нет нужды.

Виджет в окне программы

Создание виджета в окне *Qt* тоже не представляет собой ничего особенно нового, за исключением одного момента, о котором будет сказано ниже. Вот как мы создаем виджет (это фрагмент файла **dialcontrol.cpp**):

```
QDeclarativeView *qmlView = new QDeclarativeView;
dial = new Dial();
qmlView->rootContext()->setContextProperty("Dial", dial);
qmlView->setSource(QUrl("qrc:/Dial/Dial.qml"));
QVBoxLayout *layout = new QVBoxLayout(this);
layout->addWidget(qmlView);
```

Вы, наверное, сразу обратили внимание на то, какую ссылку мы используем для загрузки исходного текста модуля QML. В прошлый раз мы использовали ссылку на файл Linux. Это давало нам огромную свободу в плане модификации внешнего вида окна нашей программы, но делало ее зависимой от расположения файлов QML. Теперь мы поступаем иначе и включаем файл **Dial.qml** и все сопутствующие ему файлы в модуль ресурсов *Qt*.

В результате описание виджета QML станет частью программы *Qt*, и нам уже не придется беспокоиться о том, где хранятся соответствующие файлы. Обычно в модули ресурсов включают пиктограммы и элементы интернационализации приложения, но ничто не мешает нам включить в них модули QML, тем более что движок *Qt* QML умеет работать с пространством ресурсов приложения (и с другими пространствами URL) как с локальной файловой системой. Так что если включенному в модуль ресурсов модулю QML понадобится файл, тоже включенный в модуль ресурсов, модуль QML сможет без проблем загрузить его. Главное, чтобы ссылки на файлы были относительными, а не абсолютными.

Чтобы превратить модуль QML в ресурс программы *Qt*, нам понадобится файл описания ресурсов ***.qrc**. Работать с файлами с расширением **qrc** можно многими способами – с помощью дизайнера графических интерфейсов *Qt* или с помощью программы

» **Пропустили номер?** Узнайте на с. 107, как получить его прямо сейчас.

Qt Creator, но можно обойтись и текстовым редактором, ведь формат QRC основан на XML. Вот как, например, выглядит файл QRC для нашей программы (файл d**ialcontrol.qrc**):

```
<!DOCTYPE RCC><RCC version="1.0">
<qresource>
<file>Dial/background.png</file>
<file>Dial/Dial.qml</file>
<file>Dial/DialControl.qrc</file>
<file>Dial/needle.png</file>
<file>Dial/needle_shadow.png</file>
<file>Dial/overlay.png</file>
</qresource>
</RCC>
```

Внутри тэга <file> мы просто указываем путь к файлу ресурса относительно расположения файла QRC. Теперь при сборке приложения виджет QML будет включен в нашу программу. В результате наша программа больше не зависит от расположения файлов QML, но хорошо это или плохо?

С одной стороны, это упрощает установку программы: вам не нужно думать о том, где должны быть размещены файлы QML в соответствии со стандартом XDG, а если вы пишете кросс-платформенное приложение, ваша жизнь упрощается еще больше. С другой стороны, при таком подходе теряется одно из важнейших преимуществ QML как средства описания интерфейсов программ *Qt*: возможность радикально сменить интерфейс без повторной сборки приложения.

Существует еще своего рода компромиссный вариант: скомпилировать виджет QML как внешний двоичный ресурс программы *Qt* (для этого служит утилита *rcc*). С одной стороны, работать с внешним файлом ресурса в программе *Qt* не намного сложнее, чем со встроенным; с другой стороны, внешний файл ресурса может быть заменен без повторной сборки программы. Однако и у этого подхода есть свой минус: дизайнеру интерфейсов придется иметь дело со специальными инструментами *Qt*, такими как *rcc*, тогда как в случае расположения виджета QML целиком в собственных файлах дизайнеру для изменения интерфейса будет достаточно текстового редактора и редактора *GIMP* (или, на худой конец, *Photoshop*). В общем, наиболее разумный выбор зависит от конкретных целей – хотите ли вы, чтобы каждый пользователь, освоивший QML и растровую графику, мог «сшить новую одежку» для вашей программы, или нет.

Еще о методе setContextProperty()

До сих пор мы использовали метод setContextProperty() для передачи указателя на созданные нами объекты со свойствами. Но его возможности гораздо шире. Прежде всего, отметим, что существует и другой вариант метода setContextProperty():

```
void        setContextProperty ( const QString & name, const
QVariant & value )
```

То есть, со свойством контекста QML можно связать не только объект, но и любое значение, приводимое к типу QVariant. Например:

```
qmlView->rootContext()->setContextProperty("HelloText",
trUtf8("Hello World!");
```

Объекты, которые мы передаем контекстам, могут экспортировать в QML не только свойства, но и методы. Чтобы модуль QML увидел метод объекта, этот метод должен быть объявлен в разделе public slots. Например, еслидобавить в класс Dial метод

```
public slots:
int angleToPos(int angle)
{
…
}
```

то в файле **Dial.qml** можно написать так:

```
property int angleToPos : dial.angleToPos(angle)
```

Сбор информации об ошибках

Поскольку наша сложная конструкция QML может нуждаться в отладке, в программе предусмотрен способ вывода информации об ошибках, которые возникают в ходе выполнения программы на QML. Вообще-то в ОС Linux все сообщения об ошибках QML выводятся в стандартный поток вывода той консоли, с которой была запущена программа. В ОС Windows *Qt* этого почему-то не делает: даже если запустить программу из окна командной строки Windows, никаких сообщений об ошибках мы не увидим (даже если ошибки есть). В любом случае, наше приложение графическое, и его не обязательно будут запускать из окна консоли, так что неплохо обзавестись собственным методом вывода информации об ошибках, основанном на графическом интерфейсе.

Список актуальных ошибок возвращается методом errors() QML-виджета в виде объекта

```
QList<QDeclarativeError>;
```

Объект класса QDeclarativeError включает несколько свойств и методов, из которых мы воспользуемся методом toString(). Этот метод возвращает информацию об ошибке на человеческом языке в переменной QString:

```
errors = qmlView->errors();
for(int i = 0; i < errors.count(); i++)
MessageBox::critical(this, "QML Error", errors.at(i).toString());
```

где errors – объект вышеуказанного класса, производного от Qlist.

Вы, наверное, обратили внимание: я написал, что метод errors() возвращает информацию об актуальных ошибках, то есть о тех, которые существуют на момент его вызова. Тем, кто привык к компилируемым языкам программирования, это может быть непривычно, но в динамически исполняемых программах QML ошибки могут возникать и исчезать динамически, и не каждая ошибка обязательно приводит к аварийному завершению программы.

В связи с этим полезно сравнить вывод программы на консоль Linux (куда поступает информация обо всех ошибках) с нашим выводом с помощью класса MessageBox. Таким образом, например, я узнал, что связывать корневой контекст с объектом dial лучше до загрузки текста программы QML, ведь выполнение программы начинается немедленно после вызова метода setSource(). Если сразу после вызова setSource() программа на QML не сможет инициализировать свой объект Dial, это вызовет ошибку, которая, однако будет устранена, как только мы укажем программе, чем именно следует инициализировать этот объект. Иначе говоря, инициализировать корневой контекст можно и до вызова setSource(), и после этого вызова, но первый способ работает чище.

Последние штрихи

Последнее, что мы сделаем для того, чтобы наш виджет выглядел, как любой другой виджет *Qt* – удалим белый фон, заботливо созданный для нас объектом QDeclarativeView (не забывайте, что это потомок QGraphicsView). Мы сделаем это так:

```
qmlView->setBackgroundRole(QPalette::Background);
```

Поскольку наш виджет только показывает значения, но не позволяет их вводить, нам следует добавить в окно программы еще один виджет, предназначенный для управления виджетом QML. Как и в исходном примере программы QML, мы воспользуемся для этого ползунком, только в нашем случае это будет объект класса QSlider. Наш виджет будет реагировать на сигнал sliderChanged() этого объекта. Можно было бы добавить в класс Dial слот и связать этот слот с сигналом sliderChanged() напрямую; я оставляю вам это в качестве домашнего задания.

В следующий раз мы, наконец, рассмотрим программу на чистом QML (для выполнения которой все равно требуется утилита *Qt*). LXF

C# и Mono:

Андрей Кузьменко продолжает рассказ о переносе кода с C++ на C#. В этой статье речь пойдет о множественном наследовании.



Наш эксперт

Андрей Кузьменко Убежденный сторонник надежного ПО и любитель C++. Из всех дистрибутивов Linux отдает предпочтение Knoppix.

Сложилось так, что язык C# не поддерживает множественное наследование. Споры о том, хорошо это или плохо, продолжаются уже долгое время, однако по большей части они принимают вид «религиозной войны» и носят скорее теоретический характер. С другой стороны, при объектно-ориентированном подходе к проектированию программного обеспечения множественное наследование часто оказывается механизмом, наиболее адекватно описывающим моделируемый объект и его поведение. Кроме того, может возникнуть необходимость выполнить реинжиниринг существующего программного обеспечения: например, перенести код, написанный на **C++** с использованием множественного наследования, на платформу Mono. В данной статье будет рассмотрен один из возможных вариантов подхода к реализации подобия «настоящего» множественного наследования в языке C#. Хочу сразу предупредить читателя о том, что все те возможности в том виде, как они есть в языке C++, в рамках платформы Mono получить нельзя; однако можно провести адаптацию имеющихся языковых средств и использовать интересные и полезные практические результаты для решения определённого класса задач – а именно, объединения разнородных иерархий классов.

Встать, суд идёт

Множественное наследование позволяет создать производный класс на основе нескольких базовых классов. При этом критики традиционно указывают на две возможные проблемы. Первая заключается в том, что появляется возможность унаследовать одно и тоже же имя поля данных или метода от нескольких базовых классов, что может послужить причиной неоднозначности:

```
// Родительские классы
class First { int function(int x); };
class Second { int function(int x); };
// Дочерний класс
class Result : First, Second { }
// создан объект класса
Result res = new Result();
// какой именно метод будет вызван?
res.function(10);
```

Вторая проблема – это «ромбовидное» наследование, когда в иерархии от базового класса к производному существует более одного пути. Вот классический пример:

```
class File{ string file_name};
class InputFile : File {};
class OutputFile : File {};
class IOFile : InputFile, OutputFile {};
```

Возникает вопрос: должны ли поля данных базового класса дублироваться в объекте подкласса столько раз, сколько существует путей между ними в иерархии наследования? В языках, поддерживающих множественное наследование, таких как C++, Eiffel, Python, это решается по-разному.

Справедливости ради замечу, что список трудностей, возникающих при множественном наследовании, не исчерпывается теми двумя, что описаны выше; однако существуют ситуации, когда множественное наследование действительно оправдано – например, объединение независимых иерархий классов, композиция интерфейсов, создание класса из интерфейса и реализации. Наличие механизма множественного наследования делает язык програмирования выразительнее и богаче, и наиболее справедливый путь – это дать выбор программисту: использовать имеющиеся возможности или нет.

Теория

Если класс рассматривается как механизм для представления некоторых сущностей, то интерфейсы можно понимать как описание некоторых действий над этими сущностями. По сути своей, интерфейсы в языке C# очень похожи на виртуальные методы абстрактного класса в языке C++. Они описывают группу связанных функциональных возможностей, которые могут принадлежать любому классу и иметь методы, свойства, события, индексаторы или любое их сочетание. Интерфейсы не могут содержать поля данных. Кроме того, они не содержат реализации методов. Когда говорят, что класс наследует интерфейс, это означает, что класс предоставляет реализацию для всех членов, определяемых интерфейсом. Таким образом, в интерфейсе мы заявляем, что хотим сделать, но не определяем, как это будет конкретно реализовано. Смысл в том, что для разных классов мы реализовываем однотипный набор методов с одним видом их вызова, но при этом реализация методов будет различаться в разных классах. Допускается наследование произвольному числу интерфейсов.

> «Класс рассматривается как механизм представления сущностей.»

Существующая в языке C# замена «полноценного» множественного наследования классов на наследование интерфейсов выглядит следующим образом: есть базовый класс, который имеет некоторый набор данных и методов. Есть один или несколько интерфейсов, которые предполагают выполнение классом некоторых операций. Производный класс наследует «полноценному» классу, получая от него некоторые данные и методы, плюс принимает на себя обязательства по реализации тех интерфейсов, которым он наследует.

Понятие интерфейса легко иллюстрируется простым житейским примером. Представьте себе множество предметов, относящихся к бытовой электронике: видеомагнитофон, DVD-проигрыватель, CD-плейер, автомагнитолу. Произведённые разными фирмами-изготовителями и выполняющие разные функции, они все имеют унифицированный пользовательский интерфейс: если нажать на кнопку с нарисованным треугольником, устройство будет «играть», а если нажать кнопку с квадратом, воспроизведение прекратится.

Стереоэффект

Практика

Давайте рассмотрим программную модель мобильного телефона, который может использоваться как MP3-плейер, для работы с которым пользователю достаточно нажимать соотвествующие кнопки на корпусе.

```
using System;
namespace Gadget
{
        class MobilePhone
        {
                public void make_call(string number)
{
Console.WriteLine("Call:" + number);
}
        }

        interface IManagement
        {
                void play( );
                void stop( );
        }

        class Gadget : MobilePhone, IManagement
        {
                public void usb_connect( )
{
Console.WriteLine("Connect to computer…");
}
                public void play( ) { Console.
WriteLine("Music playing now!");}
                public void stop( ) { Console.WriteLine("Stop
playing music."); }

        }

        class Program
        {
                public static void Main(string[] args)
                {
                        Gadget g = new Gadget();
                        g.play();
                        g.stop();
                        g.make_call("89991234567");
                }
        }
}
```

Страшная тайна

В литературе, будь то учебники для начинающих или справочники для профессионалов, ответ на вопрос о том, почему же в языке C# нет «нормального» множественного наследования для классов, зачастую очень расплывчат. Дескать, множественное наследование сложно для понимания и является источником потенциальных ошибок. Я думаю, что стоит приподнять завесу таинственности и разобраться в этой ситуации.

Выше вы уже прочли, что возможно унаследовать одно и то же имя поля данных или метода от нескольких базовых классов. Как же тут быть? Язык C++ предлагает такое решение: если нет необходимости дублировать информацию в производном классе, то все непосредственные потомки базового класса в ромбовидной схеме должны наследовать своему предку виртуально:

```
class File{};
class InputFile    : public virtual File {};
class OutputFile : public virtual File {};
class IOFile       : public InputFile, OutputFile {};
```

Таким образом, File становится виртуальным базовым классом. Этот вариант, будучи простым и вполне понятным для программиста, требует определённых «трудозатрат» со стороны компилятора. Проблема в том, что виртуальные базовые классы реализуются как указатели. При этом размер кода увеличивается, доступ к полям данных виртуальных базовых классов оказывается медленнее по сравнению с невиртуальными базовыми классами, а сам компилятор, который всё это реализует, становится «тяжелее». Важное замечание: к моменту определения классов InputFile и OutputFile нет никакой информации о том, будет ли когда-нибудь какой-либо класс наследовать от них обоих или нет. Если изначально не объявить класс File виртуальным базовым классом, то может сложиться так, что разработчику класса IOFile потребуется переопределить классы InputFile и OutputFile, а это бывает невозможным по причине того, например, что данные классы находятся в скомпилированной библиотеке.

При этом правила, согласно которым происходит инициализация виртуальных базовых классов, оказываются сложнее, чем в случае «простого» наследования. При невиртуальном наследовании аргументы конструктора базового класса задаются в списке инициализации непосредственного производного класса – таким образом, аргументы передаются совершенно очевидным способом: классы уровня N транслируют аргументы вверх, классам уровня (N–1). А в случае виртуального наследования ответственность за инициализацию базового класса ложится на самый последний дочерний класс в иерархии. По этой причине классы, наследующие виртуальному базовому и нуждающиеся в инициализации, должны знать обо всех своих виртуальных предках. Кроме того, при добавлении в иерархию нового производного класса он обязан принять на себя обязательство по инициализации виртуальных базовых классов. Ко всему прочему, возникает резонный вопрос о способах корректной реализации операций копирования и присваивания в свете вышеописанных проблем. Единственный радикальный выход в сложившейся ситуации – это отказ от полей данных в родительских классах. Таким образом устраняется необходимость в передаче аргументов конструкторам виртуальных базовых классов.

Надеюсь, теперь ясности стало больше. Желающие разобраться в этом вопросе ещё глубже могут целенаправленно продолжить свои поиски, но перед этим мы получим максимум от возможностей платформы Mono. Как именно? Читаем дальше! »

Что хотим

Какую функциональность нам бы хотелось иметь, пусть даже в первом приближении? Итак:

» Дочерний класс, наследуя своим родителям, должен иметь возможность вызывать их методы без необходимости для программиста повторно переписывать код. Свобода использования того, что уже есть.

» Мы хотим иметь возможность использовать объекты дочернего класса там, где ожидаются объекты родительских классов. Это позволит нам оперировать производным классом через ссылки на базовые классы. По сути, обычный полиморфизм.

» Родительские классы для реализации механизма наследования не должны подвергаться никаким изменениям. Должен соблюдаться принцип целостности. При соблюдении данного требования мы сможем использовать родительские классы только посредством их интерфейса, что очень актуально при работе с динамическими библиотеками, когда исходный код недоступен.

Моделирование

В качестве предметной области мы будем использовать офисную технику: принтеры, сканеры и многофункциональные устройства (МФУ). Думаю, нет смысла рассказывать что такое «принтер» и «сканер». Что касается МФУ, то это устройства, прежде всего, объединяющие в себе возможности принтера и сканера, так сказать, «два в одном». Кроме того, в некоторых моделях может присутствовать функциональность факса и кард-ридера. МФУ позволяет экономить место на столе и интерфейсные разъёмы, а если присутствует поддержка операционной системы Linux, так и вообще замечательный аппарат!

Принтеры и сканеры в нашем случае символизируют две независимые иерархии, которые мы хотим объединить с целью получения новой сущности – МФУ. Очевидно, что класс принтеров обладает своими специфическими данными и методами, а класс сканеров – своими. Класс МФУ, наследуя классу «принтер» и классу «сканер», получит определённый набор методов от родительских классов, и кроме того, у него будут свои собственные поля данные и методы.

Для программной модели из всего многообразия свойств принтеров мы будем использовать свойство «язык описания страниц» (Page Description Language – PDL), а сканер будет определяться оптическим разрешением. В прилагаемом файле **LibDev.cs** содержится код классов Printer и Scaner. Типы данных, представляющие собой родительские классы Printer и Scaner, очень просты. Они вынесены в своё собственное пространство имён OfficeDevices. Данный файл компилируется в dll-библиотеку, которая затем используется в основном проекте.

Тип данных «ПРИНТЕР» содержит поле данных page_language, свойство Page_Language с процедурами доступа get и set, конструктор с параметром для инициализации поля данных и метод ShowPageLanguageInfo(), выводящий на консоль значение поля данных. Тип данных «СКАНЕР» – поле данных resolution, свойство Resolution с процедурами доступа get и set, конструктор с параметром для инициализации поля данных и метод ShowResolutionInfo(), выводящий на консоль значение поля данных.

Класс-наследник – MFU (сущность «МНОГОФУНКЦИОНАЛЬНОЕ УСТРОЙСТВО»). Именно он является нашей конечной целью.

Языковые средства

Композиция – это отношение между классами, возникающее тогда, когда объект одного типа содержит в себе объекты других типов. В качестве синонимов термина «композиция» выступают «вложение» или «агрегирование». Пример:

```
class Address {}
class ContactInfo{}
class Person
{
            Address address;
            ContactInfo contact;
}
```

Когда мы говорим о наследовании, то для нас это означает, что «класс является разновидностью другого класса». В термин «композиция» вкладывается смысл «содержит» или «реализуется посредством».

Бывает так, что при разработке программ требуется выполнить приведение одного созданного программистом типа данных к другому. Для решения такой задачи в языке C# предусмотрены средства явного и неявного преобразования типов. Если в программе есть класс First и класс Second и нам нужно выполнять неявное преобразование типа Second к типу First, то в классе First нужно объявить метод приведения следующего вида:

```
public static implicit operator First(Second s) { // необходимые действия}.
```

Фактически, выполняется перегрузка оператора.

Кодирование

Шаг первый – создание классов-наследников Scaner_tmp и Printer_tmp.

```
public class Printer_tmp : Printer
{
            internal MFU mfu_part;
    public Printer_tmp(string data) : base(data) { }
    static public implicit operator MFU(Printer_tmp p)
    {
            return p.mfu_part;
            }
}

public class Scaner_tmp : Scaner
{
internal MFU mfu_part;
    public Scaner_tmp(string data) : base(data) { }
    static public implicit operator MFU(Scaner_tmp s)
    {
        return s.mfu_part;
    }
}
```

Благодаря наследованию классов Printer и Scaner через них можно получить доступ к функциям ShowPageLanguageInfo() и ShowResolutionInfo(), а также к свойствам родительских классов. Мы видим, что каждый из этих классов-наследников имеет поле данных mfu_part и оператор для приведения типа, который позволяет преобразовать эти классы к типу MFU. Зачем это нужно? Для того, чтобы мог выполняться следующий код:

```
Printer p1 = new Printer("PostScript");
Printer p2 = new Printer("PCL6");
MFU    p3 = new MFU("PCL5", "1200x1200", "USB 1.1");

Printer [] arr = {p1, p2, p3};
foreach (Printer print_i in arr) print_i.ShowPageLanguageInfo();

Printer_tmp tmp = (Printer_tmp)arr[2];
MFU mfu = (MFU)tmp;
mfu.Mfu_SelfTest();
mfu.ShowPageLanguageInfo();
```

» **Пропустили номер?** Узнайте на с. 107, как получить его прямо сейчас.

В массиве arr типа Printer содержится три элемента: два из них – это указатели на «настоящие» принтеры, а третий элемент – указатель на объект класса MFU, работа с которым ведётся через указатель на родительский класс Printer. В цикле foreach все элементы массива «считаются» принтерами, однако в какой-то момент времени нам может понадобиться работать с третим элементом именно как с объектом класса MFU, коим он, собственно, и является.

Шаг второй: создание дочернего класса MFU.

```
public class MFU
{
Printer_tmp printer_part;
    Scaner_tmp scaner_part;
    string interface_type;

    public string Interface
    {
            get
        {
            return interface_type;
        }
        set
        {
            interface_type = value;
        }
    }
            public string Language
    {
            get
        {
            return printer_part.Page_Language;
        }
        set
        {
            printer_part.Page_Language = value;
        }

    }
    public string Resolution
    {
        get
        {
            return scaner_part.Resolution;
        }
        set
        {
            scaner_part.Resolution = value;
        }
    }
    public MFU(string lang, string res, string iface)
    {
            printer_part = new Printer_tmp(lang);
        scaner_part  = new Scaner_tmp(res);
        printer_part.mfu_part = this;
        scaner_part.mfu_part = this;
        interface_type = iface;
    }

    public void ShowPageLanguageInfo()
    {
        printer_part.ShowPageLanguageInfo( );
    }

    public void ShowResolutionInfo()
    {
            scaner_part.ShowResolutionInfo( );
    }
    public void Mfu_SelfTest()
    {
                    Console.WriteLine(“MFU ready to work!”);
    }
    static public implicit operator Scaner(MFU mfu)
    {
        return mfu.scaner_part;
    }
    static public implicit operator Printer(MFU mfu)
    {
        return mfu.printer_part;
    }
  }
```

Поля данных: interface_type – способ подключения устройства к компьютеру. Далее идут printer_part и scaner_part – это то самое агрегирование, которое поможет нам сделать подобие множественного наследования. Интерфейсы для работы с полями данных класса: Interface, Language и Resolution. Методы: ShowPageLanguageInfo() и ShowResolutionInfo() организуют доступ к методам родительских классов. Операторы преобразования типов приводят тип MFU к типам Scaner и Printer. Именно благодаря им станет возможен код вида:

```
Printer p = new MFU(“PostScript”, “600x600”, “LAN”);
Scaner s = new MFU(“PCL6”, “1200X1200”, “USB 2.0”);
```

Теперь внимательно рассмотрим конструктор класса MFU. Для «связи с предками», т. е. реализации стратегии агрегирования, создаются объекты Printer_tmp и Scaner_tmp. С их помощью мы получим доступ к функциям ShowPageLanguageInfo() и ЫhowResolutionInfo(), находящимся в классах Printer и Scaner.

Затем у созданных объектов инициализируются поля mfu_part, ссылкой на создаваемый объект типа MFU. Это будет использоваться при операциях приведения типа.

Наконец, инициализируется поле данных interface_type.

Посмотрим, как можно использовать полученный код:

```
static void Main(string[] args)
{
            Printer printer1 = new Printer(“PostScript”);
    Printer printer2 = new Printer(“SPL”);
    MFU     printer3 = new MFU(“PCL6”, “1200x1200”, “USB 2.0”);

    List<Printer> data = new List<Printer>();
    data.Add(printer1);
    data.Add(printer2);
    data.Add(printer3);

    foreach (Printer print_i in data)
       print_i.ShowPageLanguageInfo();
}
Вывод на консоль:
Page language:PostScript
Page language:SPL
Page language:PCL6
```

На этом всё. Желаю успешной работы! LXF

Mono: Подключаем библиотеки

Пусть наш файл с именем **ProgramLib.cs** содержит некий исходный код на языке C#, который мы хотим превратить в динамически подключаемую библиотеку. Файл **Program.cs** содержит код программы, который в своей работе использует код библиотеки **ProgramLib.dll**. Набираем в консоли следующие команды:

```
$ gmcs -t:library ProgramLib.cs
$ gmcs -r:ProgramLib.dll Program.cs
```

Вот таким нехитрым образом обеспечивается работа с динамическими библиотеками в командной строке.

Erlang: Язык для

Андрей Ушаков знакомит вас с концепцией функционального программирования и одним из многочисленных языков на основе этой концепции.





Наш эксперт

Андрей Ушаков Активно приближает тот день, когда функциональные языки станут мейнстримом.

Функциональное программирование звучит как академическая дисциплина в ВУЗе: нечто заумное и оторванное от реальной жизни. Еще бы не сложиться такому впечатлению – стоит только посмотреть на языки программирования из мейнстрима: C, C++, Java, C#, ... И какой из этих языков программирования функциональный? Но стоит только присмотреться повнимательней, как вы увидите элементы функционального программирования и в этих языках (по крайней мере, в большинстве). Так, например, в функциональном программировании есть концепция функций высшего порядка (более подробно см. далее) – это функции, которые принимают функции в качестве параметра, либо возвращаемое значение есть функция. Но подобные функции высшего порядка у нас встречаются и в императивных языках: так, в программах на C, C++ мы можем принимать либо возвращать указатель на функцию; в программах на C++, Java, C# мы можем принимать либо возвращать интерфейс с одним методом, и т.д.

> «Функциональный подход более абстрактен, чем императивный.»

Более того, многие императивные языки вводят элементы, поддерживающие функциональное программирование: например, в C# для подобной поддержки были введены делегаты и лямбды. В настоящее время появились и развиваются функциональные языки, которые можно отнести к мейнстриму: F#, Scala, ... Данная статья посвящена обзору одного из таких функциональных языков – Erlang, который можно описать следующей формулой: функциональный язык + процессы.

Элементы программирования

Функциональное программирование – очень большая тема, которой посвящно много книг. Давайте коротко рассмотрим особенности, преимущества и недостатки функциональных языков (желающие изучить эту тему более подробно могут обратиться к соответствующей литературе). Что же отличает функциональное программирование от императивного (от программирования, например, на языке C)? Главным образом то, как трактуется процесс вычисления: вычисление есть вычисление значений функций в математическом понимании последних. А это означает следующее: значение функции определяется только значениями переданных ей аргументов, и нет никакого внешнего состояния, от которого может зависеть значение функции.

Этот подход сильно отличается от подхода императивного программирования, в котором вычисление трактуется как последовательность изменений состояния программы. И, в отличие от функционального подхода, вычисление в императивном программировании может иметь побочные эффекты, то есть значение функции, например, зависит уже не только от значения аргументов, но и от внешнего состояния. Понятно, что если бы в функциональных языках были только вычисления, не зависящие от внешнего состояния, они вряд ли бы когда-либо вышли за рамки академического интереса. Пользовательский ввод и вывод, работа с файлами, с базами данных и т.п. – все это вычисления, зависящие от внешнего состояния (и с некоторыми побочными эффектами). И во всех функциональных языках есть те или иные средства для работы с такими вычислениями (так, в Erlang просто не все функции являются независимыми от внешнего состояния).

В чем же плюсы функционального подхода? В первую очередь, в повышении надежности кода: если нет внешнего состояния и побочных эффектов, мы легко можем предсказать результат работы программы. Отсутствие внешнего состояния и побочных эффектов позволяет нам писать более простые и надежные модульные тесты, что опять же повышает надежность кода. В многопоточной среде вычисления, которые не зависят от внешнего состояния и не имеют побочных эффектов, легко могут быть распараллелены.

Теперь поговорим о минусах. Связаны они с тем, что функциональный подход более абстрактен, чем императивный, и реализации функциональных языков вынуждены достаточно много делать «под капотом», что приводит к дополнительным расходам ресурсов компьютера, в некоторых случаях достаточно существенным. В первую очередь это необходимость наличия сборки мусора, во вторую – поддержка «ленивых» вычислений.

Стоит упомянуть еще пару концепций, которые широко используются в функциональном программировании: функции высших

процессов

порядков и анонимные функции (лямбда-выражения). Функции высшего порядка – это функции, которые принимают в качестве аргументов одну или несколько функций, либо возвращаемое значение есть функция. Так, функция, вычисляющая значение определенного интеграла, является примером функции высшего порядка, принимающей в качестве аргумента другую функцию; а функция, возвращающая функцию – решение дифференциального уравнения является примером функции высшего порядка, возвращаемое значение которой есть функция. Анонимная функция – это функция, определенная только в том месте, где она используется. Поэтому, соответственно, анонимная функция обходится без имени и не участвует в процедуре поиска функции по имени в момент компиляции.

Особенности языка

Рассмотрим некоторые элементы функционального программирования применительно к языку Erlang.

Переменные являются настолько фундаментальными объектами, что они присутствуют, пожалуй, во всех языках. И Erlang в этом плане не исключение – переменные в нем есть. Erlang является языком со строгой динамической типизацией. Поэтому при объявлении переменной ее тип не задается (в отличие, например, от C), а динамически выводится во время выполнения. При объявлении переменной очень важно, чтобы ее имя начиналось с заглавной буквы – это ее отличает от других объектов. Например, **SomeVariable** – это переменная. Значение переменной может быть присвоено один и только один раз, и после присвоения ее значение не может быть изменено. Таким образом, мы всегда знаем, в каком месте программы переменная получила свое значение, и это очень сильно упрощает отладку.

Присваивая значение переменной, мы используем оператор = (например, **SomeVariable = 1**). Но этот оператор не является оператором присвоения (что очень неожиданно для человека, имеющего опыт программирования на императивных языках вроде C): это оператор соответствия [pattern matching]. При вычислении выражения **Left = Right** происходит проверка, соответствуют ли **Left** и **Right** друг другу. **Left** и **Right** будут соответствовать друг другу в следующих случаях:

» **Left** и **Right** имеют одно и тоже значение одного и того же типа. Например, **1 = 1**.

» Объявляется переменная **Left**. В этом случае, т. к. переменная **Left** никакого значения не имеет, ей присваивается значение **Right**.

Операция соответствия появляется не только при использовании оператора =. В любом месте программы, где возможно появление операндов **Left** и **Right**, будет происходить проверка на их соответствие. Например, пусть у нас есть следующее объявление функции (более подробно о функциях см. далее):

```
f(0) -> true;
f(Number) -> f(Number-1).
```

В первой строке содержится конкретизирующий вариант функции **f()**. При вызове функции будет происходить поиск в порядке объявления вариантов. При этом заданное значение параметра будет проверяться на соответствие значению, определенному в варианте. Так, например, вызов **f(0)** приведет к вызову первого варианта, а вызов **f(1)** – к вызову второго, причем параметру **Number** будет присвоено значение **1**.

Что будет, если при проверке двух операндов **Left** и **Right** выявится, что они не соответствуют друг другу? Если мы использовали оператор =, то получим ошибку времени выполнения. Если это было, например, при просмотре очередного варианта объявления функции, то поиск перейдет к следующему варианту.

Типы данных

Давайте поговорим теперь о типах данных Erlang. Наиболее очевидные типы – это целые и действительные числа (было бы странно, если бы их не было в языке программирования). Другой, чуть менее очевидный тип данных – это атомы. Аналог атомов – константы, но, в отличие от констант, атомы не содержат связанных с ними значений. Объявление атомов всегда начинается с маленькой буквы: например, **atom_sample**. »

Другие функциональные языки

» **Scala** – мультипарадигмальный язык, спроектированный кратким и типобезопасным для простого и быстрого программирования. В нем сочетаются возможности функционального и объектно-ориентированного подходов. Основной целью разработки был язык, обладающий хорошей поддержкой компонентного ПО.

» **Haskell** – стандартизованный чистый функциональный язык программирования общего назначения. Является одним из самых распространенных языков с поддержкой отложенных вычислений.

» **Lisp** – семейство языков программирования, данные в которых представляются системами линейных списков символов. Язык является функциональным, но многие поздние версии обладают также чертами императивности; к тому же, имея полноценные средства символьной обработки, становится возможным реализовать объектно-ориентированность. Наиболее популярные в наши дни диалекты языка Lisp – это Common Lisp и Scheme.

» **ML** – семейство строгих языков функционального программирования с развитой полиморфной системой типов и параметризуемыми модулями. Это не чисто функциональные языки, так как включают и императивные инструкции. ML преподается во многих западных университетах (в некоторых даже как первый язык программирования).

» **Miranda** – функциональный язык программирования, имеющий строгую полиморфную систему типов и поддерживающий типы данных пользователя. Как и язык ML, преподается во многих университетах.

» **OCaml** – современный объектно-ориентированный язык функционального программирования общего назначения, который был разработан с учетом безопасности исполнения и надежности программ. Язык OCaml поддерживает функциональную, императивную и объектно-ориентированную парадигмы программирования.

» **F#** – функциональный язык программирования общего назначения. Структура F# во многом схожа со структурой Ocaml, с той лишь разницей, что язык F# реализован поверх библиотек и среды исполнения .NET.

Следующие два типа данных являются фундаментальными в большинстве (если не во всех) функциональных языков программирования: это кортежи и списки. Кортеж [Tuple] – это контейнер для хранения объектов разных типов. Его аналогом являются, например, структуры языка C, но, в отличие от структур, поля кортежа не имеют имени. Доступ к ним можно осуществить либо по индексу поля, либо с использованием операции соответствия. Например, мы объявляем следующий кортеж: **TupleSample = {1, atom_sample}**. Тогда доступ к первому полю осуществляется следующим образом: **element(1, TupleSample)**, а для доступа к первому полю через операцию соответствия мы используем следующий синтаксис: **{First, Second} = TupleSample**, после чего переменная **First** содержит значение из первого поля кортежа **TupleSample**. Список – это контейнер для хранения объектов одного типа. Аналогами списка являются, например, массивы языка C, класс ArrayList языка Java. Список объявляется следующим образом: **[Element1, Element2, ...]**, например, **[1, 2, 3]**. Существует специальный синтаксис для доступа к головному и хвостовому элементам списка через операцию соответствия: **[Head | Other]** и **[Other | Tail]**, где **Other** – оставшаяся часть списка. Для создания списков также существует специальная методика, аналогов которой в императивных языках программирования нет – List Comprehensions. Не углубляясь во все возможности этого мощного механизма, просто приведем пару примеров.

» Генерация списка квадратов чисел от 1 до 10:

```
[N*N || N <- lists:seq(1, 10)].
```

» Генерация списка квадратов чисел от 1 до 10 только для четных чисел:

```
[N*N || N <- lists:seq(1, 10), N rem 2 == 0].
```

А как же строки, спросите вы. Неужели в Erlang сделали шаг назад в отношении работы со строковыми данными? В Erlang отдельного типа данных для строк нет: строки представляются в виде списков целых чисел. Но существует «синтаксический сахар» для работы со строками, позволяющий записывать строковые данные в привычном виде. Например, литерал "123" синтаксически корректен и соответствует значению [49, 50, 51]. К сожалению, в привычном для нас виде могут быть записаны строки только в кодировке Latin-1 (ISO-8859-1). Поэтому в других кодировках необходимо работать напрямую с данными в списках.

Помимо перечисленных здесь типов данных, в языке Erlang существуют и другие типы. Наиболее интересный и значимый из них – это тип двоичных данных. Но углубляться в эти типы данных мы не будем.

Функции

Пожалуй, последнее и самое важное, что следует рассказать об Erlang – это объявление функций. В самом простом случае, функция объявляется обычным образом: имя функции (оно должно начинаться с маленькой буквы), после которого следует список аргументов (через запятую) в скобках (для объявления аргументов действуют те же правила именования, что и для переменных), а за ним следует тело функции. Например, объявление функции, вычисляющей квадрат своего аргумента, будет выглядеть так:

```
square(Number) -> Number*Number.
```

Функции (сигнатуры функций, если быть более точным) различаются не только по имени, но и по числу аргументов, которые они принимают (арность функции). Поэтому мы можем иметь две или более функции с одинаковым именем, но разным числом аргументов.

Но на этом возможности определения функции не заканчиваются. Мы можем определить два и более варианта одной и той же функции. При вызове функции будет происходить последовательный просмотр всех объявленных вариантов (в порядке их объявления) и поиск первого подходящего. Какой вариант будет подходящим, зависит от переданных аргументов. Если же не будет найдено ни одного подходящего варианта, мы получим ошибку времени выполнения.

> «Какой вариант будет подходящим, зависит от аргументов.»

Благодаря каким механизмам варианты функции будут отличаться друг от друга? Таких механизмов два: операция соответствия и охранные выражения [guards]. Механизм соответствия действует следующим образом: при объявлении варианта функции для одного или нескольких аргументов мы вместо самих аргументов задаем какие-либо значения. И при поиске подходящего варианта эти значения будут сравниваться со значениями аргументов, а если между ними будет соответствие, то будет выбран этот вариант функции. Например, мы можем объявить функцию для вычисления факториала следующим образом:

```
factorial(0) -> 1;
factorial(1) -> 1;
factorial(N) -> N*factorial(N-1).
```

Другой механизм – это охранные выражения. Охранное выражение – это логическое выражение с аргументами функции. Если при поиске подходящего варианта охранное выражение истинно, то выбирается этот вариант. Охранное выражение объявляется при помощи ключевого слова **when**. Например, функцию для вычисления факториала мы можем объявить при помощи охранного выражения следующим образом:

```
factorial(N) when N == 0 or N == 1 -> 1;
factorial(N) -> N*factorial(N-1).
```

Понятно, что при объявлении функции у нас могут быть варианты, использующие соответствие, или использующие охранные выражения, или использующие и то, и другое.

С функциями связано еще одно фундаментальное понятие – хвостовая рекурсия. Если при выполнении функции последней операцией будет вызов самой себя (с теми же либо с другими аргументами), то этот вызов будет не рекурсивным (этот вызов будет представлять собой простой переход на начало функции). Если же вызов самой себя будет располагаться где-то внутри тела (не будет последней операцией), то это будет обычный рекурсивный вызов. Так, например, в функции **factorial()** будет хвостовая рекурсия, а в функции **factorial2()** – нет:

```
factorial(N) when N == 0 or N == 1 -> 1;
factorial(N) -> N*factorial(N-1).
factorial2(N) when N == 0 or N == 1 -> 1;
factorial2(N) →
        N*factorial(N-1),
        log(N).
```

Хвостовая рекурсия важна тем, что она позволяет организовать цикл (конечный или бесконечный) и не приводит к заполнению стека (которое возникает при обычном рекурсивном вызове).

Пример

Хватит теории – давайте решим простую задачу для демонстрации некоторых концепций, изложенных выше. В качестве примера возьмем задачу номер 17 с проекта Project Euler (http://projecteuler.net/index.php?section=problems&id=17). Условие этой задачи звучит следующим образом:

Если числа от 1 до 5 записать английскими словами (one, two, three, four, five), то будет использовано 3 + 3 + 5 + 4 + 4 = 19 букв. Если все число от 1 до 1000 включительно записать английскими словами то сколько будет использовано букв?

Примечание: Пробелы и дефисы при подсчете не учитывать. Например, 342 (three hundred and forty-two) содержит 23 буквы, 115 (one hundred and fifteen) – 20 букв. Использование союза "and" соответствует британскому варианту.

Нашей задачей будет написать программу, которая считает количество букв, необходимое для записи английскими словами чисел от 1 до N, при условии, что N не превышает 1000. По правилам британского варианта английского языка, между сотнями и десятками (если таковые есть) ставится артикль "and" (например, числу 115 соответствует строка one hundred and fifteen). Поэтому для удобства мы разделим обработку десятков и сотен в нашей программе. Метод **less_hundred/4** обрабатывает числа, которые меньше 100. В этом методе мы подсчитываем количество символов, необходимых для записи числа, меньшего 100, английскими словами:

```
less_hundred(Number, _, _, _) when Number >= 100 ->
erlang:error(badarg);
less_hundred(Number, From0To9, _, _) when Number =< 9 ->
        length(lists:nth(Number+1, From0To9));
less_hundred(Number, _, From10To19, _) when (Number > 9) and
(Number < 20) ->
        length(lists:nth(Number-9, From10To19));
less_hundred(Number, From0To9, _, OtherTens) ->
        length(lists:nth((Number div 10)-1, OtherTens)) +
length(lists:nth((Number rem 10)+1, From0To9)).
```

Обратите внимание, как мы определяем несколько вариантов функции **less_hundred/4** при помощи охранных выражений.

Следующий шаг – обработка чисел, не превышающих 1000. Для этого служит метод **parse_number/4**:

```
parse_number(1000, _, _, _) -> length("one" ++ "thousand");
parse_number(Number, From0To9, From10To19, OtherTens) when
Number >= 100 ->
LessHundred = less_hundred(Number rem 100, From0To9,
From10To19, OtherTens),
        if
                LessHundred == 0 ->
length(lists:nth((Number div 100)+1, From0To9)) +
length("hundred");
                LessHundred /= 0 -> length(lists:nth((Number
div 100)+1, From0To9)) + length("hundred" ++ "and") +
LessHundred
        end;
parse_number(Number, From0To9, From10To19, OtherTens) ->
        less_hundred(Number rem 100, From0To9, From10To19,
OtherTens).
```

Обратите внимание, что при определении нескольких вариантов функции **parse_number/4** мы используем как механизм соответствия, так и охранные выражения. Следует сказать, что в данном методе мы применяем конструкцию, которую в обзоре не обсуждали – конструкцию **if**. Эта конструкция состоит из последовательности пар выражений (разделенных оператором **->**) и возвращает некоторое значение. Возвращаемое значение определяется следующим образом: последовательно просматриваются пары выражений, в каждой паре вычисляется значение первого выражения, и если значение первого выражения равно **true**, то значение второго выражения становится возвращаемым (либо будет брошена ошибка времени выполнения, если поиск завершится не успешно).

Осталось сделать совсем немного – обработать все числа от 1 до некоторого максимально числа (не превышающего 1000).

```
parse_numbers(MaxNumber, From0To9, From10To19, OtherTens)
->
        parse_numbers(MaxNumber, 1, 0, From0To9,
From10To19, OtherTens).

parse_numbers(MaxNumber, CurrentNumber, LetterCount, _, _, _)
        when CurrentNumber > MaxNumber -> LetterCount;
parse_numbers(MaxNumber, CurrentNumber, LetterCount,
From0To9, From10To19, OtherTens) ->
        parse_numbers(MaxNumber, CurrentNumber +
1, LetterCount + parse_number(CurrentNumber, From0To9,
From10To19, OtherTens), From0To9, From10To19, OtherTens).
```

Для такой обработки мы вводим две функции: **parse_number/4** и **parse_number/6**. Эти функции являются разными за счет того, что у них разное количество аргументов (разная арность) – это вариант перегрузки функций в Erlang. Про эти функции следует заметить следующее: **parse_number/4** является интерфейсной функцией к **parse_number/6**, а **parse_number/6**, соответственно, функцией реализации. Это удобный и широко применяемый в Erlang подход, когда функция с меньшим числом аргументов объявляется для удобства использования, а функция с тем же самым именем и большим числом аргументов выполняет всю работу (подобные примеры интерфейсных объектов и объектов реализаций можно легко найти в императивных языках).

Ну и, наконец, экспортируемая функция, для взаимодействия с нашей функциональностью (**solve/1**):

```
solve(MaxNumber) when MaxNumber > 1000 ->
erlang:error(badarg);
solve(MaxNumber) ->
        From0To9 = ["", "one", "two", "three", "four", "five",
"six", "seven", "eight", "nine"],
        From10To19 = ["ten", "eleven", "twelve", "thirteen",
"fourteen", "fifteen", "sixteen", "seventeen", "eighteen", "nineteen"],
        OtherTens = ["twenty", "thirty", "forty", "fifty", "sixty",
"seventy", "eighty", "ninety"],
        parse_numbers(MaxNumber, From0To9, From10To19,
OtherTens).
```

Осталось только привести объявления модуля и экспортируемых функций:

```
-module(problem_017).
-export([solve/1]).
```

Вот и все решение задачи. Запускаем в консоли среду выполнения (запуском исполняемого файла **erl**; надеюсь, пути у вас настроены), в консоли Erlang запускаем сначала компиляцию **c(problem_017).**, а потом и выполнение нашей программы **problem_017:solve(1000)**, и получаем следующий результат: 21124. Если зайти на сайт проекта Project Euler, в раздел задачи номер 17, то можно убедиться, что это правильный ответ (http://projecteuler.net/index.php?section=problems&id=17).

О многозадачности и взаимодействии между процессами мы поговорим в следующей части этого учебника. LXF

Полезные ссылки и книги

- http://www.erlang.org/ – главный сайт (с документацией и исходным кодом среды).
- http://www.trapexit.org/ – сайт Erlang-сообщества (форум, вики, решения, учебные пособия, справочные материалы).
- http://erlanger.ru/ – сайт русского Erlang-сообщества.
- http://groups.google.com/group/erlang-russian – русское Erlang-сообщество на Google.
- http://www.tryerlang.org/ – онлайн-интерпретатор Erlang.
- Martin Logan, Eric Merritt, and Richard Carlsson "*Erlang and OTP in Action*".
- Francesco Cesarini, Simon Thompson "*Erlang Programming A Concurrent Approach to Software Development*".
- Joe Armstrong "*Programming Erlang: Software for a Concurrent World*".

KDevelop: Среда

Грэм Моррисон объединяет два самых горячих релиза из мира KDE с *Qt*, чтобы создать сияющий новизной текстовый редактор.





Наш эксперт

Грэм Моррисон Редактор наших обзоров, а иногда и разработчик KDE, экономит время, применяя регулярные выражения, чтобы потратить его на возню со старыми синтезаторами.

KDevelop – среда разработки приложений KDE. Она стоит на двух столпах KDE – C++ и API *Qt*, но в ней можно писать программы и на других языках, включая PHP. Впрочем, вас может удивить, что мы снова к ней вернулись – ведь в **LXF134** мы довольно резко высказались о только что вышедшем первом релизе *KDevelop 4.0*, и основания у нас на это были. Так что же изменилось?

Ну, на тот момент *Qt Creator*, новая «родная» среда разработки для программистов *Qt*, оставляла *KDevelop* далеко позади. Имея безошибочную коррекцию синтаксиса, встроенную документацию и встроенный же дизайнер пользовательских интерфейсов, *Creator* также предоставляла лучший способ получить и последние релизы *Qt*, и среду разработки кроссплатформенных проектов. Да и предоставляет. Однако за два последних релиза *KDevelop* выросла в полноценную альтернативу, особенно для разработчиков KDE, и если вы еще не слышали о новых возможностях и улучшениях стабильности в версии 4.2, самое время с ними познакомиться.

Преимущества KDevelop 4.2

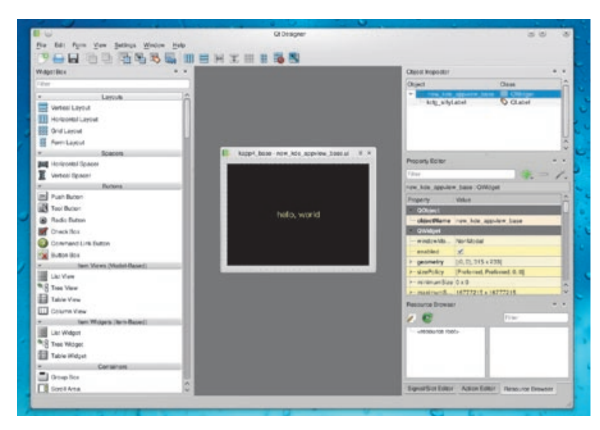
С *KDevelop 4.2* можно создавать приложения в среде разработки, которая понимает добавки KDE к API *Qt*. Здесь также есть удобная онлайн-справка, подсветка синтаксиса и возможность управления системой сборки **cmake**. *KDevelop 4.2* и KDE 4.6 вышли одновременно, а значит, *KDevelop* – единственная ныне среда разработки, включающая все возможности KDE 4.6 по умолчанию. Но начинать с ней работу непросто, особенно если вы избалованы поддержкой для новичков от *Qt Creator*. Мы покажем, что, одолев несколько «лежачих полицейских», с собственной средой разработки KDE можно работать так же продуктивно, как со средой от Nokia. Разница – в подходах к новому проекту.

Системные требования

Чтобы компилировать программы в *KDevelop*, нужна среда разработки. Также вам понадобятся библиотеки разработчика KDE (пакеты с расширением DEV) и все их зависимости. Четвертое поколение KDE требует дополнить старый инструментарий **make** утилитой **cmake**, поэтому установите и ее. Для проверки, что **cmake** установлена и работает, наберите **cmake** в командной строке, и то же самое стоит попробовать сделать с пакетом исходных кодов KDE 4, чтобы убедиться, что все установлено и готово к сборке. Загрузите, например, последнюю версию *Amarok* и скомпилируйте ее с помощью **cmake**.

Покончив с настройкой системы, запускайте *KDevelop*. На фоне *Qt Creator* он выглядит голым: есть только простой мастер запуска и, в отличие от *Creator*, вспомогательных средств для начала нового проекта мало. Для разработчиков KDE число шаблонов проектов, включенных в *KDevelop*, равно нулю. Здесь даже нельзя наскоро создать стартовый проект в качестве отправной точки.

Однако не все потеряно: вам не придется набирать каждую строчку простой программы KDE вручную, чтобы получить базовый шаблон и заставить его работать. Эту работу за вас сделает одна из утилит KDE, и благодаря волшебству **cmake** вы сможете импортировать начальную структуру проекта в *KDevelop* с той же легкостью, как если бы проект был создан внутри самой среды. Итак, создайте себе копию *KAppTemplate*, мастера запуска, которого



› ***Designer* от Nokia остается самой удобной средой для быстрой разработки интерфейсов пользователя ваших приложений KDE.**

» **Месяц назад** Мы бороздили просторы Интернета с web-сервером *Cherokee*.

для разработки

нет в *KDevelop*. Лучше воспользуйтесь версией из репозитория пакетов своего дистрибутива, а не той, что в Интернете, потому что домашняя страница проекта не обновлялась несколько лет.

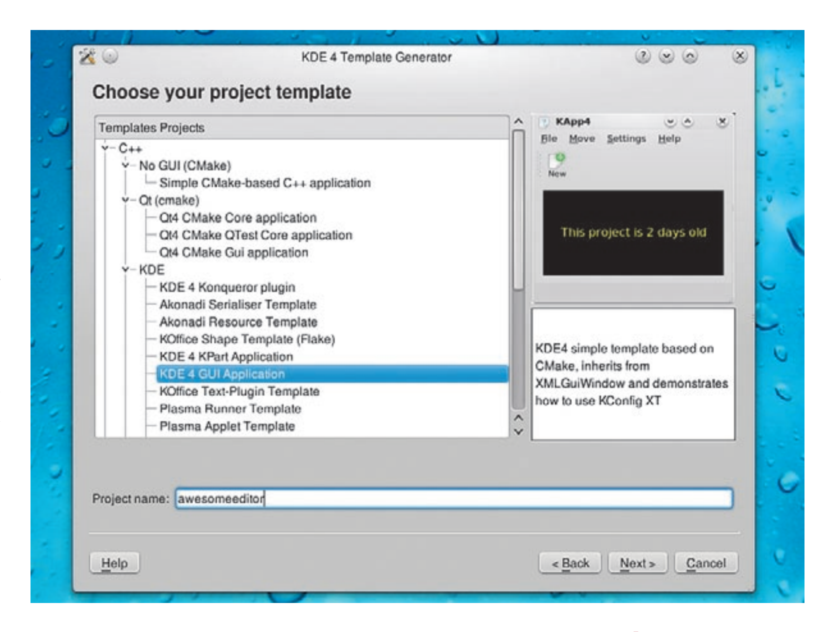
Разобраться в самом приложении очень легко. Самый важный шаг – второй: тут вы задаете тип создаваемого проекта. В нашем релизе перечислены 10 различных типов проектов KDE, от модулей расширения *Konqueror* до апплетов *Plasma*. Тот, что нам интересен – KDE 4 GUI Application, базовый шаблон для большинства стандартных приложений KDE. Задайте имя проекта и затем, на третьем шаге, создайте файлы и сохраните их в новый каталог.

Теперь импортируйте этот проект в среду *KDevelop*. Закройте мастер запуска *KDevelop*, если он появился, и выберите Open/Import Project [Открыть/Импортировать Проект] в меню Project [Проект]. С помощью кнопки выбора файлов найдите каталог свежесозданного проекта и выберите файл **CMakeLists.txt**, который сгенерировался в корневом каталоге проекта **root**. Если вы не видите его, убедитесь, что текстовые файлы есть в фильтре, и нажмите на кнопку Next [Далее]. Этот файл содержит все зависимости проекта, и *KDevelop* сумеет преобразовать этот проект в свой собственный формат. На последнем этапе, импорте, убедитесь, что в качестве системы сборки [Build System] выбран CMake Project Manager, и нажмите на кнопку Finish [Конец]. Появится другое окно, где нужно будет указать расположение файлов **cmake**, но если они уже настроены, значения этих полей можно не менять.

Ранние сборки

Внешний вид *KDevelop* со времен третьей версии серьезно переделан, и хотя экран, как правило, заполнен довольно плотно, в основных областях окна разобраться легко, если вам случалось иметь дело с другой средой разработки. Например, слева находятся файлы проекта и классы, содержащиеся в этих файлах. Также можно открыть сразу несколько проектов, поэтому перед компиляцией стоит убедиться, что выбран верный проект и его расположение указано правильно. Справа вы найдете окошко для просмотра шаблонов кода, точнее, небольших часто используемых фрагментов кода. Например, можно занести туда оператор ‘for’ или ‘if’ и вставлять их в текущую позицию курсора одним нажатием кнопки, а не набирать эти конструкции вручную с риском допустить ошибки. Шаблоны можно даже отсортировать по категории: например, по языку программирования. Этот компонент взят из модуля расширения *Kate*, предоставляющего те же возможности.

Прежде чем двигаться дальше, скомпилируйте новый проект, чтобы убедиться, что все работает. Для этого нажмите F8 или кнопку Build Selection [Скомпилировать выбранный проект] на панели инструментов, предварительно убедившись, что в дереве проектов [Projects] выбран корневой каталог проекта. Когда начнется процесс компиляции, вы увидите, что панель вывода внизу изменилась, и теперь на ней выводится результат работы команды **cmake**. Там отображаются обнаруженные ошибки, и если они вызваны неправильным кодом, можно щелкнуть по ним и попасть в соответствующую строку файла исходного кода. Это стандартная практика в средах разработки, подобных *KDevelop*, но так как никакого кода мы еще не написали, проблем быть не должно.



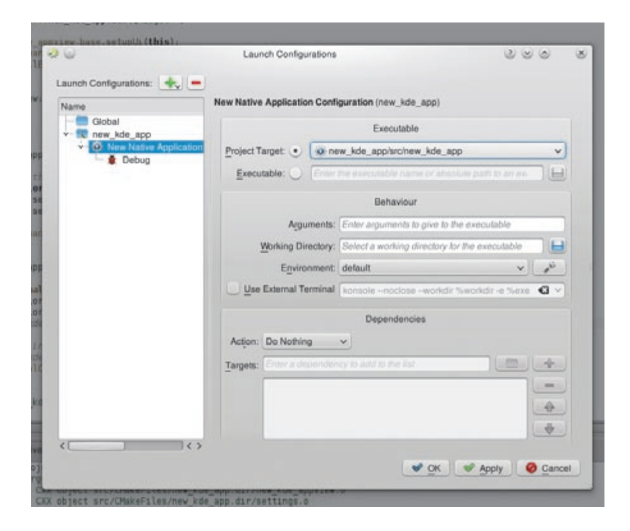
› *KappTemplate* сгенерирует вам львиную долю кода стартового проекта. Смело пишите разработчикам благодарственные письма...

После компиляции *KDevelop* еще не знает, где найти исполняемый файл. Нажмите F9 или нажмите на кнопку Execute (Выполнить) и выберите свой проект из списка – обычно это второй проект из двух предложенных. Щелкните на плюсике, чтобы изменить настройки проекта, и выберите двоичный файл из выпадающего списка Project Target [Цель проекта]. Если он не появляется, введите его расположение вручную в виде **appname/src/appname**. Нажмите Apply [Применить], и файл должен запуститься. Если вам понадобится вернуться в это окно, воспользуйтесь пунктом Configure Launch Configuration [Настройка конфигурации запуска] меню Run [Запуск]. На экране появится окно приложения, а на панели вывода *KDevelop* будут отображаться все ошибки выполнения. Если все работает и вы довольны, закройте приложение.

Теперь пора добавить немного своего функционала. Самое большое различие между *KDevelop* и *Qt Creator* в том, что в первом нет встроенного дизайнера интерфейсов. Это было бы проблемой, так как в любой уважающей себя среде разработки ныне обязан быть редактор интерфейсов в том или ином виде. К счастью, на помощь приходит другое стороннее приложение. На сей раз это почтенный *Qt Designer*, самостоятельная версия того же дизайнера, что в *Qt Creator*. Он уже должен быть установлен как часть пакетов разработчика *Qt*, используемых в KDE 4, и его можно открыть, запустив исполняемый файл **designer-qt4**.

По сути, *Qt Designer* – это тот же самый компонент *Qt Creator*, но в этой версии вы должны найти не только список виджетов *Qt*, но и виджеты KDE. Они перечислены на большой панели слева, и с ее помощью виджеты можно добавить в окна своей программы. У нашего приложения-шаблона уже есть интерфейс, и, к счастью, он не жестко зашит в исходных файлах, а использует »

› **Перед запуском проекта, построенного *KDevelop*, сообщите среде разработки, где находится исполняемый файл.**



формат файла *Qt Designer* – UI. Чтобы загрузить этот файл, пропустите мастер создания новой формы, появляющийся при первом запуске приложения, и через меню Open [Открыть] перейдите в каталог с исходным кодом проекта **src**. В нем находятся две формы UI – одна для окна приложения и одна для диалога настроек [Preferences]. Нам интересен файл главного окна приложения **appview_base.ui**.

Загрузив этот файл, вы должны увидеть нечто похожее на выполняющуюся программу. Окно состоит из единственной метки внутри черного квадрата, безо всякой иронии говорящей 'hello, world'. В отличие от окончательной версии программы, вы не увидите в этом окне меню и панель инструментов. Оба эти элемента добавляются позже в исходном коде проекта. Мы уберем виджет метки и заменим его чем-нибудь более функциональным. Для этого аккуратно выделите его мышью и нажмите клавишу Delete. Приятно то, что в полном виджетов *Designer* можно свободно перетаскивать виджеты в свое приложение и удалять их оттуда, не беспокоясь об исходном коде. Например, можно добавить новый виджет KDE *WebKit*, мгновенно наделив свое приложение самым современным web-интерфейсом, или добавить список процессов из списка виджетов *KSysGuard*. После добавления в исходный код проекта информации о заголовочных файлах оба виджета будут работать без всяких дополнительных действий.

Однако в нашем проекте мы заменим виджет метки виджетом *KRichTextEdit*. Он делает именно то, о чем вы подумали, предоставляя пользователю панель редактирования текста с заданными шрифтом, размером, выделением жирным и курсивом. Впрочем, он не позволяет изменять эти параметры, чем мы и займемся далее.

Сигналы и слоты

Сохраните свою работу в *Qt Designer* и вернитесь в *KDevelop*. Прежде чем перекомпилировать приложение, чтобы увидеть наши изменения в интерфейсе, нужно убрать все следы метки, которую мы уничтожили в *Designer*. Их можно найти в файлах **appview.cpp** (ко всем используемым здесь именам файлов спереди нужно добавить название проекта). Найдите две строки, ссылающиеся на виджет *kcfg_sillyLabel*, который мы удалили с формы. Закомментируйте их, добавив в начале строки **//**, или удалите. Если теперь скомпилировать и запустить проект, вы увидите, что на месте метки появилось окно текстового редактора, и в него можно ввести текст. Щелкнув на нем правой кнопкой мыши, вы также сможете скопировать, вырезать и вставить текст, а также выполнить специфичные для KDE задачи, такие как перевод и проверку орфографии. Они зависят от используемой установки KDE, но появляются в программе сами.

Однако сами не появляются возможности, обязанные быть в приличном текстовом редакторе – а именно, выделение текста

KDevelop 4.2

Панель инструментов
Это панель инструментов общего назначения, где можно скомпилировать и запустить приложение, добавить новые файлы или выполнить поиск по проекту.

Просмотр файла или класса
Здесь выбираются файлы, которые надо отредактировать, в зависимости от того, что доступно на панели вкладки.

Панель выполнения
На этой панели можно переключиться между консолями компиляции и выполнения, командной строкой и контекстной справкой для выбранных компонентов.

Панель выбора проекта
Если открыто несколько проектов, выберите на этой панели тот, который нужно скомпилировать.

Редактор кода
Он основан на *Kate*, и многие возможности, включая сворачивание текста, проверку синтаксиса и всплывающие подсказки, у них идентичны.

Вторая панель
Все панели в *KDevelop* легко настраиваются. По умолчанию эта панель справа используется для просмотра документации и отладки.

Автодополнение кода
KDevelop 4.2 – лучшая на данный момент среда разработки с точки зрения автодополнения кода KDE. Она мигом предложит возможные варианты дополнения на основе вашего кода и API KDE или *Qt*.

жирным и курсивом и возможность изменять стиль шрифта. В редакторе есть этот функционал, но ожидается, что интерфейс реализуют сами разработчики. И в *Qt*, и в KDE это означает работу с сигналами и слотами. С их помощью компоненты в обоих средах взаимодействуют друг с другом, образуя нечто вроде трубопровода. Например, один отправляет сигнал, типа нажатия кнопки, а другой принимает это действие и отправляет его в слот, такой как выделение выбранного фрагмента жирным шрифтом в текстовом редакторе. В результате большинство приложений в KDE и *Qt* состоят из сети соединений событий с фрагментами кода, которые их обрабатывают, и описание сигналов и слотов каждого виджета занимает большую часть документации API обоих сред.

Мы добавим кнопку на панель инструментов и воспользуемся сигналом «нажато» [pressed] этой кнопки для активации одной из функций форматирования в текстовом редакторе. Панели инструментов и меню в KDE обрабатываются не так, как другие элементы интерфейса, поэтому их не видно в *Designer*. Так происходит потому, что они называются действиями, а не виджетами; их можно легко добавить, переместить и изменить их настройки. Именно поэтому, например, почти во всех приложениях KDE есть легко настраиваемая панель инструментов. Расположение иконок на панели инструментов можно найти в основном файле настройки приложения, и после того как утилита KDE установлена, этот файл будет находиться в главном каталоге с конфигурацией KDE, который зависит от дистрибутива. Из *KDevelop* этот файл можно открыть с помощью двойного щелчка на файле RC в окне Project.

В этом файле уже есть запись для действия пункта меню **Move**, и мы заменим ее, создав вместо нее действие **Bold**. Нам потребуется изменить имя поля на **bold**, и текст тоже на **Bold**, но самая важная часть этого файла – поле действия. Почему? Потому, что на это поле направлена ссылка на код из файла настройки. Поэтому вам нужно создать имя действия, которое мы также добавим и в приложение – пусть это будет **makeBold**. Сохраните файл и откройте главный исходный файл проекта (**appname.cpp**).

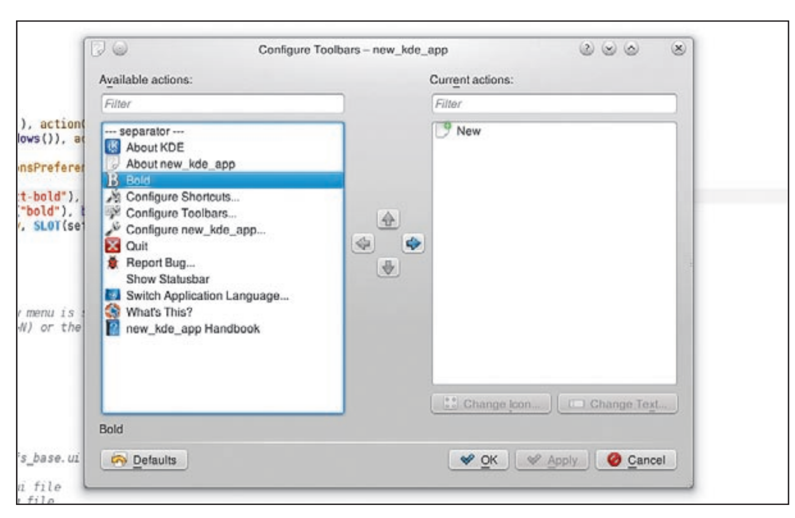
Метод в действии

Сигналы и слоты для каждого действия задаются в методе **setupActions**; это самый верхний блок кода в файле. В нем будет находиться единственное действие длиной в три строки кода, название которого начинается с **KAction**. Мы изменим его так, что вместо него будет использоваться действие **Bold**. Строка **KIcon** в первой строчке – ссылка на изображение, используемое для иконки. В KDE есть множество стандартных изображений для иконок, и, к счастью, среди них есть одно для обозначения выделения текста. Оно называется **format-text-bold**. Следующую строку в той же самой строчке нужно заменить на **bold**. Это текст, который мы увидим в приложении, и он помещен внутрь функции **i18n**, так что переводчики смогут легко локализовать текст. Этой же схемой следует пользоваться для любой строки, которую видит пользователь.

В строке, следующей за **KAction**, обе строки нужно заменить на **bold**, и первая используется для связи этого действия с названием пункта меню в конфигурационном файле. Наконец, в последней строке конфигурации мы задаем сигнал и слот, и она должна выглядеть следующим образом:

```
connect(bold, SIGNAL(triggered(bool)), m_view, SLOT(setBold()));
```

Она связывает сигнал, отправляемый пунктом меню **bold**, со слотом **setBold** в приложении. Правда, мы еще не создали слот **setBold**. Как вы видите из вызова функции connect, приложение ожидает, что этот слот будет находиться в объекте **m_view**, и это



› После добавления нового действия Bold в ваше приложение потребуется добавить функцию на панель инструментов с помощью внутренней утилиты конфигурации.

связано с кодом в заголовочном файле **appname_view** и исходных файлах. Сначала откройте заголовочный файл и добавьте директиву **#include "krichtextwidget.h"** к списку директив в начале файла, а метод **void setBold();** – в раздел приватных слотов этого метода. Эта функция будет приемником нашего сигнала, и теперь нужно добавить код обработки входящего сигнала в главный файл CCP. Откройте файл **view.cpp** и создайте в нем следующую функцию:

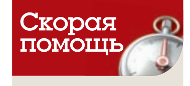
```
void appnameView::setBold()
{ ui_appnameview_base.krichtextwidget->setTextBold(true);
}
```

Автодополнение кода поможет вам при переименовании файлов приложения своего проекта, а также при поиске метода **bold** в виджете текстового редактора. Вот и все, что необходимо для того, чтобы текстовый редактор заработал, и вы видите, как легко добавить в такой редактор все остальные необходимые функции.

Впрочем, после компиляции и запуска приложения вас ждет разочарование: в панели инструментов нет ни следа новой иконки. Дело в том, что действие-то мы добавили, но не вставили его в конфигурацию панели инструментов. Щелкнув на панели инструментов правой кнопкой мыши и выбрать пункт меню Configure Toolbars [Настроить панели инструментов], вы увидите, что в списке слева есть действие **Bold**. Выбрав это действие и щелкнув по стрелке вправо, можно переместить его на активную панель инструментов.

По выходу из этого окна на панели инструментов появится наша кнопка, и ее нажатие сделает выделенный текст жирным. Если вы хотите, чтобы иконка была в начале панели инструментов, закройте программу и взгляните на файл **appui.rc**, который находится в вашем домашнем каталоге в каталоге **.kde/share/appname**. Это сохраненное состояние вашего приложения, и его можно скопировать в каталог с исполняемым файлом приложения, тем самым задав настройки и расположение компонентов приложения.

Конечно, чтобы от вашего приложения был прок, нужно сделать еще очень много, но с помощью рассмотренных нами приемов добавление нового функционала (даже неотъемлемой функции сохранения) не должно представлять для вас особых трудностей. В виджете редактора также есть метод **textOrHtml**, который выведет HTML-код для редактируемого файла, а уж переслать его во внешний файл несложно. Просто пользуйтесь документацией и подсветкой синтаксиса *KDevelop* и сообщите нам о своих успехах. LXF



KDevelop будет работать со всеми последними версиями KDE API между 4.2 и 4.6. Он также будет работать с шаблонами PHP, созданными *KappTemplate*.

» **Через месяц** Надев спасательный жилет, выручаем Linux-машины в сети.





Ответы

Есть вопрос по открытому ПО? Пишите нам по адресу answers@linuxformat.ru

В этом месяце мы ответим на вопросы про:

1 Поврежденные файлы
2 Использование демона *at*
3 Общий доступ к домашнему разделу
4 Нечуткие сенсорные панели
5 Тестирование скорости соединения
6 Странные префиксы
7 Прием сигналов DAB и GPS
8 Преобразователи USB в последовательный сигнал
9 DVD-риппинг

1 Ошибка ввода/вывода

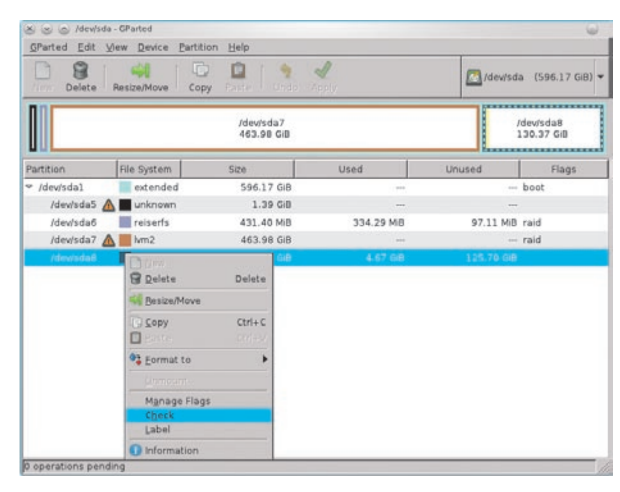
В **Мой жесткий диск объемом 1 ТБ разбит на разделы, поделенные между Windows XP и Linux Mint, плюс раздел подкачки и раздел размером 500 ГБ для резервной копии данных. А еще у меня есть второй диск 500 ГБ, где хранятся все мои данные — с ним-то и возникла проблема.**

Я только что обнаружил, что мой ежемесячный процесс резервного копирования перестал работать из-за ошибки чтения. Методом исключения я проследил причину проблемы вплоть до файла, расположенного в неком каталоге, видимом в XP, но не в Linux. В XP он невелик — только 73 байта, но мне не удалить ни его, ни папку, в которой он находится. Зарегистрировавшись в Linux как root, я уперся в ту же проблему. При попытке удалить файл появляется сообщение об ошибке:

```
Error stating file as /media/Data_in/medical/.~.pmr_ra_cts.doc
```

Запуск *Parted Magic* и *Test Disk* с целью выполнения анализа цилиндра приводит к появлению ошибки чтения read error 396/15/15, **и я вправе предполагать, что нашел причину проблемы. Я не хочу переформатировать весь диск, только чтобы освободиться от проблемного файла — не подскажете ли вы мне способ удалить этот файл без форматирования диска?**

Джон Йейтс [John Yates]



› ***GParted* умеет проверять файловые системы на ошибки, но инструменты командной строки *fsck* дают больше информации.**

О В Вашей файловой системе появился поврежденный блок, не позволяющий выполнить ни одну операцию. Файл, на который влияет этот блок, скорее всего, серьезной роли не играет. Это не сам DOC-файл, потому что они не бывают столь малы, как 73 байта. Так что это, скорее всего, файл блокировки [lock file], оставшийся после редактирования – его потеря не создаст Вам проблем.

Решение зависит от причины и серьезности повреждения файловой системы. Если оно чисто программное, что бывает из-за некорректного останова или сбоя системы, это проблема решаемая. Вы не сообщили, какая у Вас файловая система, но в случае NTFS сначала попробуйте запустить *Scandisk* в Windows. Альтернативный вариант, пригодный для любой файловой системы – применить один из инструментов *fsck*, например, *e2fsck*, если это файловая система ext2/3/4. Пусть это первый раздел на втором диске; тогда скомандуйте

```
sudo e2fsck -f /dev/sdb1
```

Файловая система при выполнении этого процесса должна быть отмонтирована (*e2fsck* известит Вас, если это не так). Кроме того, можно запустить *Gparted* с большинства Live CD – просто щелкните правой кнопкой мыши на затронутом разделе и выберите команду Check.

Если это не поможет, значит, либо файловая система повреждена очень серьезно, либо налицо сбой оборудования – сообщения об ошибках, поступающие от *fsck*, подскажут, что имеет место. В любом случае, первый шаг – перекинуть свои данные в безопасное хранилище. После этого воссоздание файловой системы и возврат данных на место не составят труда, если причина состояла в повреждении файловой системы.

Если же проблема с аппаратурой, или на диске появился поврежденный блок, то команда **format** соответствующим образом его пометит, и диск продолжит работу, лишь немного уменьшив емкость. К сожалению, поврежденные блоки похожи

Наши эксперты

» Мы найдем ответы на любой вопрос — от проблем с установкой системы или модемом до сетевого администрирования; главное — спросить!



Нейл Ботвик
Владелец ISP и экс-редактор дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Пол Хадсон
Пол – местный супер-программист, и он может и хочет управиться со всеми вашими проблемами по части web и баз данных.



Валентин Синицын
В редкие свободные минуты Валентин обычно запускает *mcedit*, чтобы отшлифовать какое-нибудь открытое приложение. Его любимая тема – настольный Linux.



Майк Сондерс
Майк был одним из создателей прототипа **LXF** – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и SNES.



Грэм Моррисон
Когда Грэм не обозревает кучи программного обеспечения и не халтурит с *MythTV*, он готов к ответам насчет любого оборудования и проблем виртуализации.



Юлия Дронова
Если компьютер у Юлии не занят выполнением команды *emerge*, она спешит применить его для модерирования www.unixforum.org.

КУДА ПОСЫЛАТЬ ВОПРОСЫ

Пишите нам по адресу answers@linuxformat.ru **или спрашивайте на форуме:** www.linuxformat.ru

на автобусы – то ни одного не дождешься часами, а то они сыплются один за другим. Кроме того, существует явная возможность того, что Ваш диск проявил симптом предстоящей кончины. Жесткие диски используют систему под названием SMART (self-monitoring, analysis, and reporting technology), способную определить ранние симптомы отказа. Активизируйте SMART в BIOS на Вашем компьютере, затем установите пакет *smartmontools* (http://smartmontools.sourceforge.net) и скомандуйте

```
sudo smartctl -t short /dev/sdb
```

Тест пойдет в фоновом режиме. Подождите, пока не получите сообщение о его завершении, и просмотрите, нет ли ошибок, по команде

```
sudo smartctl -l error /dev/sdb
```

Более подробную информацию выведет команда

```
sudo smartctl -a /dev/sdb
```

Если обнаружатся проблемы, замените диск, пока не потеряно что-нибудь важное. Кроме того, существует демон *SMART*, работающий в фоновом режиме: он наблюдает за дисками и уведомляет по электронной почте об обнаружении проблем. Сайт *smartmontools* и сопутствующая документация объяснят все это подробнее. НБ

2 At с трех попыток

В Я с большим интересом прочел вашу статью про демон *at*, и, как фанат *cron*, счел ее любопытной. Я пытаюсь запустить скрипт преобразования видео на моем третьем разделе (Storage2), но при запуске программы в 15:00 командой

```
echo "./toavi.sh" | at 15:00
```

ничего не происходит. Я ввел команду cd /Storage2/'Audio Visual'/Anime **в начало скрипта, и при запуске ее вручную все работает хорошо. Нет ли способа запустить команду** at, **перенаправив вывод в файл** logfile.txt, **чтобы узнать, в чем проблема?**

Стив Томпсон [Steve Thompson]

О Вывод команды **at** довольно скуден, но есть пара мест, куда стоит заглянуть. В первую очередь, определите, какие именно задания **at** выполняет, набрав

```
atq
```

Выведется очередь заданий от Вашего пользователя. В первом столбце указан номер задания, и им можно воспользоваться для вывода подробных сведений о задании. Следующая команда (где **N** – номер задания) выведет скрипт, который **atd** запустит в указанное время:

```
at -c N >jobscript
```

Этот длинный скрипт состоит в основном из всяких присваиваний, так как он воссоздает среду на момент времени, в который задание было запланировано, включая текущий каталог. Вы можете проверить, что произойдет, командой

```
/bin/sh jobscript
```

Если она сработает, то следующий шаг – проверить, что с ней сделает команда **at**. Если команда **at** не сможет запустить задание, она должна сделать об этом запись в системном журнале – обычно он называется **/var/log/messages**. Чтобы прочесть этот файл, надо зарегистрироваться в системе от имени root. Найдите сообщения от **atd**, дав команду:

```
sudo grep atd /var/log/
messages
```

Когда команда **atd** запустит задание, она использует **/usr/sbin/sendmail** для отправки по почте вывода, stdout и stderr пользователю, запустившему ее. Добавление параметра **-m** к команде **at** велит ей отправлять информацию по почте даже тогда, когда никакого вывода сгенерировано не было. Большинство дистрибутивов имеют локальную почтовую систему, но Ваш почтовый клиент может быть не настроен на сбор локальной почты, и многие люди часто вообще забирают почту только с почтового сервера своего ISP. Почта обычно хранится в каталогах **~/.Mail** или **~/.maildir**, в зависимости от настройки локальной почтовой системы. Настройте свой почтовый клиент на прием оттуда почты с целью просмотра вывода **atd**.

Запуская скрипт оболочки, можно повысить его информативность, чтобы увидеть, далеко ли заходит его выполнение и нет ли проблем. Некоторые любят применять в скриптах команды **echo**, но сначала добавлять их, а потом за ненадобностью удалять не слишком удобно. Более простое решение – добавить в начало скрипта

```
set +x
```

Эта команда велит *Bash* (или другой используемой Вами оболочке) выводить на экран каждую строку перед ее исполнением, и Вы будете видеть, что происходит. Сравните вывод при ручном запуске этой команды с тем, что пришлет Вам по почте **atd**, и определите, на каком этапе скрипт застопоривается.

При тестировании Вы можете предпочесть управлять выполнением задания, а не передавать его **atd**. Остановите сервис **atd**, затем выполните задание один раз с помощью:

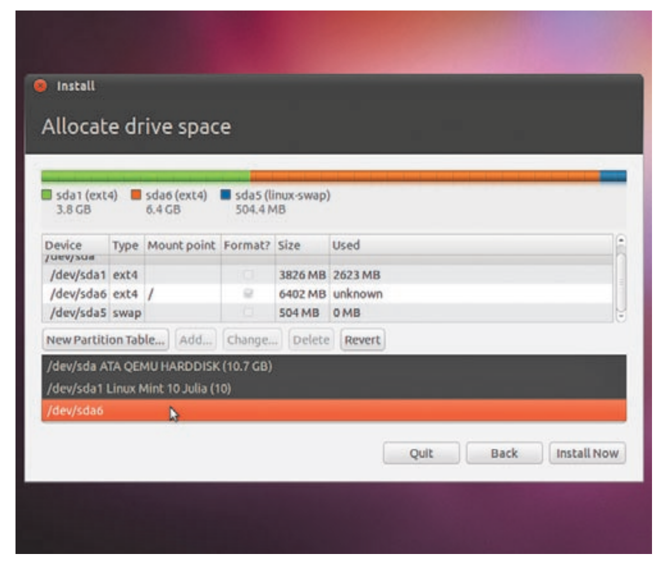
```
echo "./toavi.sh" | at now
sudo atd -s
```

Эти команды обработают все задания, поставленные в пакетную очередь, и завершат выполнение. НБ

3 Меню мультизагрузки

В Я хотел бы добавить в свою систему с двойной загрузкой еще один дистрибутив Linux. На первом жестком диске у меня Windows XP Media Centre. На втором жестком диске я создал раздел swap, раздел boot и раздел root для Linux Mint 10, а также большой раздел home. Как я понимаю, можно использовать раздел /home из обоих дистрибутивов, и, конечно, пользоваться одним и тем же разделом swap.

Джоэль А. Хаус [Joel A House]



› Устанавливая второй дистрибутив Linux, попросите инсталлятор поместить загрузчик в корневой раздел этого дистрибутива.

О Безопаснейший способ добавить в Вашу систему еще один дистрибутив (кстати, FreeBSD не является дистрибутивом Linux, но совет и для него годится), не сбивая существующий у Вас загрузчик – велеть установщику нового дистрибутива поместить свой загрузчик в раздел **root**, а не в главную загрузочную запись (MBR) первого жесткого диска. Тогда имеющаяся загрузочная конфигурация останется нетронутой.

Перезагрузившись в первый раз, Вы увидите прежнее загрузочное меню, причем без опции выбора нового дистрибутива. Не волнуйтесь, с новым загрузчиком все в порядке – Вам просто нужно известить о нем старый загрузчик. Загрузитесь в Mint, вызовите файл **/etc/grub.d/40_custom** в Вашем любимом текстовом редакторе от имени root и дайте следующие команды, где **hdX,Y** – это раздел, содержащий корень нового дистрибутива и его загрузчик:

```
menuentry "FreeBSD" {
 insmod ufs2
 set root=(hdX,Y)
 chainloader +1
}
```

Grub использует другой способ нумерации разделов: **X** — это номер диска, а **Y** – номер раздела на диске. Чтобы подчеркнуть отличие, *Grub* и отсчет начинает с нуля, так что **/dev/sdb5** превращается в **hd1,4**. Теперь скомандуйте

```
sudo update-grub
```

Изменения войдут в силу. При последующей загрузке в меню появится еще одна опция, для выбора нового дистрибутива.

При установке нового дистрибутива, использующего **Grub2**, ситуация немного отличается: здесь применятся *OS-prober*, который определяет другие ОС и добавляет их автоматически. Если Вы установите Xubuntu после Mint, то установщик обнаружит Mint и добавит в загрузочное меню. Если у Вас стоит FreeBSD, возможно, понадобится скопировать файл **/etc/grub.d/40_custom** с раздела Mint. »

Альтернатива – велеть инсталлятору Xubuntu установить загрузчик на своем корневом разделе, а затем загрузить Mint и скомандовать

```
sudo update-grub
```

Установленная копия Xubuntu будет обнаружена и добавлена в меню Mint.

Для совместного использования раздела **swap** несколькими дистрибутивами Linux достаточно велеть инсталлятору использовать существующий **swap**-раздел. Разделение **swap** между Linux и FreeBSD возможно, но требует ухищрений: потребуется отредактировать скрипты инициализации Linux, чтобы при каждой загрузке на разделе перед его монтированием запускалась команда **mkswap**. Обычно проще бывает завести для Linux и FreeBSD отдельные разделы **swap**.

Раздел **/home** совместно использовать легко: только велите инсталлятору не форматировать этот раздел, а просто воспользоваться им. Но один и тот же раздел **home** для всех дистрибутивов – плохая идея: они могут использовать разные версии одного и того же ПО, что может привести к ужасным конфликтам файлов настройки. Если Ваш пользователь везде имеет один и тот же идентификатор – а при работе с производными Ubuntu так и будет, потому что у первого пользователя всегда идентификатор 1000 – Вы можете настроить разные пользовательские каталоги для всех дистрибутивов. Простейший способ сделать это – назвать пользователей разными именами. После этого Вы сможете предоставить в общий доступ важные файлы, создав символические ссылки на каталоги из одного домашнего каталога в другие, например:

```
sudo ln -s /home/mintuser/documents /home/
xubuntuuser/
sudo ln -s /home/mintuser/photos /home/
xubuntuuser/
```

НБ

4 Тачпад в параличе

У нас с женой два ноутбука Acer Aspire. Один из них, модель 7720g, работает под Windows 7 и Ubuntu 10.10 под *Wubi*, не зная горя. Второй, модель 5553, тоже поставлялся с Windows 7, но не дает Ubuntu работать с клавиатурой и тачпадом. Инсталляция под *Wubi* проходит хорошо, но после перезагрузки начинаются проблемы. Выбор в меню *Grub* проходит нормально, но затем, перед появлением экрана регистрации, мелькает сообщение:

```
[6.663913] disabling IRQ #18
```

Загрузка продолжается, но когда появляется экран регистрации, ни клавиатура, ни тачпад не работают. Продолжить работу можно, подключив мышь через USB и пользуясь экранной клавиатурой, но мышь управляется очень плохо. Подключение USB-клавиатуры помогло больше — все клавиши, в том числе и клавиши курсора, работают нормально.

Мы искали информацию в Web по ключевым словам interrupt #18, **но решения не нашли. Не подскажете ли вы, что предпринять?**

Майкл и Шейла Бэйтс [Michael and Sheila Bates]

Похоже, это довольно частая проблема. Одно из предположений – добавить к загрузочным параметрам ядра **irqpoll**. Чтобы протестировать этот совет, при загрузке нажмите клавишу Shift и удерживайте ее, пока не появится меню *Grub*, затем выберите загрузку Ubuntu и нажмите клавишу E, чтобы отредактировать параметры загрузки. Найдите строку, начинающуюся с **linux**, перейдите в ее конец и добавьте **irqpoll**, чтобы вся строка заканчивалась примерно так:

```
quiet splash irqpoll
```

Теперь нажмите клавиши Ctrl+X для перезагрузки с новыми опциями. Если это решит проблему, можно сделать изменение постоянным, отредактировав файл **/etc/default/grub** от имени root. Из терминала скомандуйте

```
sudo nano /etc/default/grub
```

Теперь найдите настройку **GRUB_CMDLINE_LINUX_DEFAULT** и добавьте в конец строки **irqpoll**, чтобы она выглядела примерно так:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet
splash irqpoll"
```

Нажмите Ctrl+X, чтобы сохранить настройки и выйти, затем скомандуйте

```
sudo update-grub
```

Изменение войдет в силу. Теперь элемент загрузочного меню по умолчанию будет включать **irqpoll**. Кроме того, можно изменить и загрузочные параметры для Live CD. При загрузке нажмите клавишу Shift и удерживайте ее, пока не появится загрузочное меню. Затем нажмите F6, чтобы появилось меню Other Options, затем – Esc, чтобы закрыть его. Теперь Вы сможете отредактировать параметры непосредственно и нажать Enter, чтобы загрузиться.

Известна и другая причина проблемы: конфликт с IDE-контроллером. Тогда войдите в меню BIOS Вашего компьютера при загрузке – в руководстве пользователя должно быть сказано, какую клавишу нажать, чтобы попасть в это меню.

Найдите опцию для контроллера SATA и измените режим контроллера с IDE на AHCI. AHCI – стандартный режим для контроллеров SATA, но режим IDE предоставляется, и обычно используется по умолчанию, потому что некоторые программы не ладят с AHCI – в частности, ряд версий Windows. Если проблема вызвана не **irqpoll**, так Вы почти наверняка ее решите. Windows 7 должна справляться с дисками AHCI, хотя известно, что Vista имела с этим проблемы, если только не установить добавочный драйвер. Симптом этой проблемы – жалоба Windows на необнаружение

Часто задаваемые вопросы

DHCP

» Что такое DHCP?

Это сокращение от dynamic host configuration protocol [динамический протокол настройки хостов]. Подключая компьютер к сети, нужно дать ему определенную информацию – например, IP-адрес, куда отправлять запросы для компьютеров, о которых он не знает, и адреса DNS-серверов. Вместо того, чтобы делать все это вручную, что обычно называют статической адресацией, компьютер может воспользоваться динамической информацией, получаемой с сервера. Это облегчает работу с большими сетями и упрощает перемещение между сетями – как часто бывает с ноутбуками.

» DNS? А это что такое?

Доменная служба имен [domain name service]. Это сервис, преобразующий доменные имена, как, например, www.linuxformat.co.uk, в сетевые – например, 212.113.202.71.

» Как это работает?

Клиент, то есть ваш компьютер, рассылает запрос «обнаружения» через сеть. Принимают запрос все компьютеры сети, а отвечают на него только серверы DHCP. Все серверы возвращают свои «предложения»; клиент выбирает, какое из них принять, и запрашивает у сервера соответствующую информацию – IP-адрес для компьютера, адреса DNS-серверов и адрес шлюза.

» Шлюза? В смысле, Gateway? Изготовителя компьютеров?

Нет, это другой шлюз. Шлюзом по умолчанию в сети называется адрес, на который посылается весь трафик для компьютеров вне вашей сети. Шлюз отправляет трафик во внешнюю сеть и маршрутизирует к вам ответы.

» Где мне найти сервер DHCP?

Обычно для этого не нужен отдельный пакет. Большинство маршрутизаторов, включая широкополосные или беспроводные модемы/маршрутизаторы, содержат DHCP-сервер, предоставляющий адреса для вашей локальной сети. Провайдеры доступа через кабельные модемы тоже используют DHCP для предоставления вам IP-адреса, когда вы включаете свой компьютер и модем. Если вы хотите запустить собственный DHCP-сервер для своей сети, то практически наверняка ваш дистрибутив включает и серверную, и клиентскую части DHCP. Поищите DHCP через свой менеджер пакетов или воспользуйтесь *dnsmasq* с www.thekelleys.org.uk/dnsmasq.

» А что такое периоды DHCP?

Когда вы закрываете сетевое соединение, ваш компьютер извещает сервер DHCP, что адрес освободился, и сервер может использовать его повторно. Если это не сделано должным образом, IP-адрес серверу не возвратится. Периодом [lease] DHCP называется срок, на который сервер DHCP выдает клиентам IP-адреса во временное пользование – ваш клиент знает, что до истечения этого срока надо запросить о продлении адреса, или его отберут и выдадут другому клиенту.



› Web-интерфейс маршрутизатора настроит DHCP на выдачу конкретных адресов конкретным машинам.

загрузочного диска. Если Вам нужно активизировать в Windows драйверы AHCI, см. http://support.microsoft.com/kb/922976. **НБ**

5 Беспроводной оптимизм

В У меня Sony Vaio с беспроводной картой Intel PRO/Wireless 3915ABG (802.11a/b/g). Недавно я обновил свой дистрибутив до Ubuntu 10.10 и, к моему разочарованию, скорости моих беспроводных соединений катастрофически упали. В моей системе с двойной загрузкой я в Windows 7 получаю скорости до 40 MBps, а в Ubuntu не больше 3 MBps. Я пытался найти ответ на форумах и даже послал отчет о проблеме в Canonical, но безуспешно. С момента обновления до 10.10 прошло уже два месяца, но ничего не изменилось. Что за дела?

Марк Джезинки [Mark Jesinkey]

О Похоже, что Windows Вас надувает. Предельная скорость 802.11g – 54 Mbps; здесь буква b строчная, и речь идет о мегабитах. Так как в байте 8 бит, теоретический максимум скорости соединения – 6,75 MBps. В реальности, накладные расходы TCP, необходимость и принимать, и передавать пакеты, а также неидеальный прием еще снижают эту цифру, чуть ли не вдвое. Скорость в 40 MBps требует соединения Gigabit Ethernet. Вы не привели данные о скорости в более старой версии Ubuntu, но вряд ли она была много выше.

Есть очень удобный инструмент для тестирования «сырой» скорости [raw speed, скорость с учетом передачи служебной информации, – *прим. пер.*] сетевого канала: *iperf* (http://iperf.sourceforge.net), доступный через репозитории Ubuntu. Установите его на двух компьютерах в Вашей сети. Это терминальная программа, но она очень проста – нужно запустить ее как сервер на одном из компьютеров командой

```
iperf -s
```

Затем на втором компьютере запустите программу как клиента, командой

```
iperf -c HOSTNAME
```

Здесь **HOSTNAME** – либо имя хоста, либо IP-адрес компьютера, где работает сервер. Программа проработает 10 секунд и сообщит скорость передачи данных в мегабитах в секунду для обоих терминалов. Несколько опций позволяют модифицировать вывод: например, опция **-f M** переключает отображение вывода на мегабайты. Это «сырой» трафик TCP, и фактически передачи файлов будут идти на более низкой скорости, но полученные результаты дадут представление о качестве Вашего канала связи.

Для проверки я протестировал ноутбук через канал связи 802.11n, поставив его в метре от точки доступа, и получил скорость 4,5 MBps, а для канала Gigabit Ethernet – 6,75 MBps, так что 40 MBps для беспроводного канала – это чистая фикция.

Программа *iperf* может не только протестировать Ваш канал связи, но и замерить влияние внесенных изменений. Вас удивит, как сильно меняется ситуация при простом перемещении относительно точки беспроводного доступа.



› **И в новом, и в старом *Grub* можно редактировать опции загрузки «на лету», нажав клавишу E.**

6 Про слэш и Bash

В Работая с командной строкой, я допустил мелкую ошибку, которая до сих пор меня интригует. Я не смог найти этому обоснования, но, может, я что-нибудь проглядел. Я переходил в корневой каталог моего Debian командой

```
$ cd /
```

но мои пальцы перестарались, набрав

```
$ cd //
```

Вместо выдачи сообщения об ошибке, подсказка и pwd показали //, а не /. Дивясь, не появилась ли некая новая ветвь в моей файловой системе, я вывел ее содержимое и обнаружил, что и в /, и в // все одинаково.

Похоже, что или *Bash*, или файловая система съедают двойной слэш в начале пути, но какой в этом смысл? Я уже заметил его использование для URL, например, таких, как ftp://some.server или http://other.sites, но при чем тут мой корневой каталог? Бывают ли случаи, когда полезен локальный путь с двойным слэшем?

Лоран [Laurent]

О Видимо, это идиосинкразия *Bash*, применимая только к двойному слэшу в начале пути. Команда **cd //mnt** ведет себя так же, но команды **cd ///** или **cd / mnt//cdrom** – нет. Двойной слэш производит впечатление безвредного косметического ляпа, который существует довольно давно. Поиск в Google показал, что похожий вопрос уже задавался шесть лет назад. Он применим только к *Bash* – остальные оболочки, например, *Zsh*, ведут себя так:

```
% cd //
% pwd
/
```

Это поведение ожидаемо. Оно не имеет ничего общего с двойным слэшем в URL, где:// служит разделителем между названием протокола (**http**, **ftp** и т. д.) и именем хоста.

У Вас есть три варианта действий: игнорировать эту ошибку, перейти на другую оболочку или аккуратнее печатать. **НБ**

7 DAB или GPS

В Существуют ли интерфейсы — либо карты PCI, либо USB-брелки — через которые на ПК с Linux можно было бы принимать радио DAB (настоящий сигнал DAB, а не станции, передаваемые через DVB, которые можно принимать на TV-тюнер) и сигнал GPS?

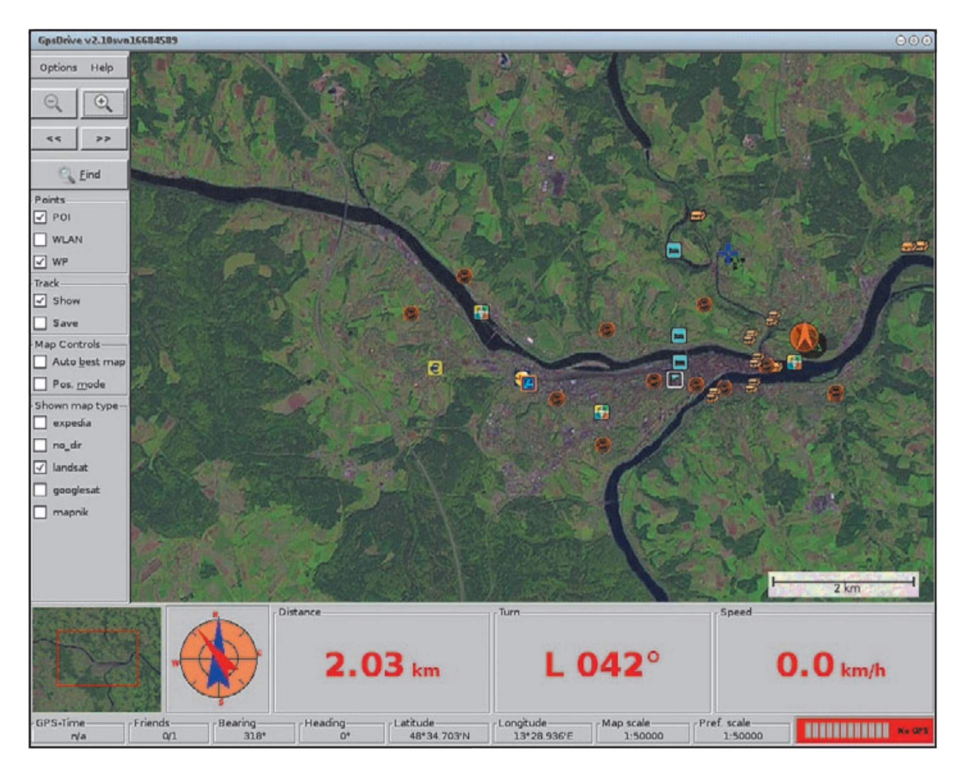
Эндрю Вуд [Andrew Wood]

О Существуют как PCI-, так и USB-приемники, и ядро Linux включает драйверы для типового USB-приемника. Правда, найти их довольно сложно. Возможно, Вам больше повезет с приемом Ваших любимых станций через Internet, например, через мобильный 3G-модем, потому что у них обычно тот же битрейт и качество.

С GPS ситуация получше, так как существует ряд самостоятельных приемников GPS для связи через Bluetooth. Я когда-то пользовался таким и получал неплохие результаты. Соединение Bluetooth избавляет от проводов между приемником и компьютером – хотя приемник надо разместить там, где есть хорошая видимость неба. Приемники GPS применяют для связи стандартный протокол NMEA 0183, так что любой из них должен работать, и все они предоставляют к ПО интерфейс последовательного типа.

Основная программа GPS для Linux – *GpsDrive* (www.gpsdrive.de), она отображает Ваше местоположение на масштабируемой карте. Если Вы хотите замкнуть приемник на свое собственное ПО, »

демон *Gpsd* (http://gpsd.berlios.de) предоставит интерфейс и диагностические инструменты. Настройка ПО зависит от Вашей аппаратуры и среды, но оба сайта предоставляют обильную документацию. Не зная, зачем Вам это нужно, более детальный совет дать трудно. **НБ**



› ***GpsDrive*** **предоставляет масштабируемую карту для любого Linux-компьютера, соединенного с приемником GPS.**

8 Последовательный USB

В Я хочу подключить мой старый наладонник 200LX Palmtop к ноутбуку ThinkPad T60, но у него нет последовательного порта. Как лучше всего это сделать? Наладонник использует последовательный кабель для подключения к настольному ПК через последовательный порт COM1.

jozien17, с форумов

О Вам нужен переходник-адаптер с USB на последовательный порт; он и стоит недорого. Они делаются для различных целей – от бесперебойных источников питания до контроллеров X10 и счетчиков электроэнергии – и работают так же, как и сам последовательный интерфейс. Все мои адаптеры были в разное время куплены на eBay – и часто это лучшее место для таких покупок, потому что в розничных магазинах электроники цены обычно выше, чем в онлайн-магазинах.

Существует масса чипсетов для этих устройств, и большинство, если не все, поддерживаются ядром Linux. Для подстраховки покупайте устройства, у которых особо упомянута поддержка Linux – меньше будет мороки. Да и зачем делать бизнес людям, которые Linux не признают?

Будучи подключены, эти устройства создают устройство ttyUSB – поэтому первое будет называться **/dev/ttyUSB0**. Для «чистых» последовательных портов ситуация будет иной, потому что Linux-эквивалент COM1 – **/dev/ttyS0**. Изменив в своем ПО настройки так, чтобы они отражали это отличие, Вы не натолкнетесь ни на какие проблемы.

Единственной возможной сложностью может оказаться отсутствие корректного модуля, если у Вас индивидуальное ядро. После подключения устройства запустите **dmesg** или скормите Google номера идентификаторов, полученных от команды **lsusb**, чтобы выяснить, что это за устройство и какой драйвер ему нужен. Любое стандартное ядро типового дистрибутива должно иметь их по умолчанию. **НБ**

Терминалы и суперпользователи

Мы часто предлагаем решение в виде команд, вводимых в терминале. Хотя обычно то же самое можно выполнить и через графические инструменты, входящие в состав дистрибутива, отличия между этими инструментами означают, что такие решения будут узко специфичными. Команды терминала гораздо более гибкие и работают во всех дистрибутивах.

Команды настройки системы часто должны запускаться от имени суперпользователя (superuser или root). В зависимости от вашего дистрибутива, есть два способа сделать это. Многие дистрибутивы – в частности, Ubuntu и его производные – требуют предварять такие команды префиксом **sudo**. После этого пользователю предлагается ввести пароль, и система выдает ему полномочия root, действующие на время выполнения команды. Другие дистрибутивы используют префикс **su**, применение которого требует ввода пароля root и дает полные права root до тех пор, пока пользователь не введет команду **logout**. Если ваш дистрибутив использует **su**, выполните эту команду один раз, и потом сможете выполнять любую команду, не предваряя ее префиксом **sudo**.

Краткая справка про...

Псевдонимы

Автозавершение экономит время набора команд и имен файлов, но еще нужно помнить или отыскивать правильные аргументы. Не проще ли установить их по умолчанию? А может, вообще наплодить команд запуска программы с разными аргументами? Все это, и еще многое, возможно при использовании псевдонимов [alias, алиасов] оболочки.

Скорее всего, по умолчанию псевдонимы у вас уже есть. Наберите '**ll**' в оболочке, и, вероятно, увидите список файлов в длинном формате, с размером файлов, датой и владельцами – потому что многие дистрибутивы делают **ll** псевдонимом **ls -l**. Чтобы увидеть доступные псевдонимы, наберите **alias** без аргументов.

Для создания псевдонима, наберите:

```
alias la="ls -lhA --color=auto"
```

Теперь, набрав **la** [list all, показать все] в текущей оболочке, вы увидите список с цветным выделением, подробной информацией и отображением скрытых файлов. Введенный вами псевдоним заменяется соответствующей ему строкой перед выполнением его оболочкой. Но при открытии новой оболочки вашего псевдонима может там и не быть: в каждой оболочке работает своя команда alias. На вид здесь работы еще больше, чем с запоминанием команд, но ее можно автоматизировать. К некоторым файлам оболочка обращается при каждом запуске. Общие для всей системы настройки содержатся в **/etc/profile**, там-то дистрибутив и хранит свои псевдонимы. Затем каждый пользователь может настраивать собственные псевдонимы в файлах конфигурации своей оболочки.

Псевдоним умеет выполнять больше одной команды, например:

```
alias foobar="foo --foo-opts ; bar --bar-opts"
```

Так можно автоматизировать простые последовательности команд, не прибегая к скриптописанию.

Большой вопрос Какой лучший способ DVD-риппинга?

В Я хочу списать содержимое моих DVD в файлы, максимально сохранив качество. На это у меня есть две причины. Во-первых, мне нужен доступ к моим фильмам через видеоплейер *MythTV*, чтоб не перелопачивать диски, если захотелось посмотреть кино. Во-вторых, я хочу смотреть фильмы на нетбуке, когда я в пути.

Как лучше это сделать? GUI-программа будет кстати, но мне бы хватило и утилиты командной строки, только без ввода сотен загадочных опций.

Брайан Донован [Brian Donovan]

О «Лучший» способ зависит от компромиссов, на которые Вы готовы пойти. Лучшее качество и самое полное воспроизведение функций DVD дает зеркальное копирование DVD. Компромиссом в данном случае является то, что на каждый диск потребуется 9 ГБ памяти на жестком диске. Вы можете создать ISO-образ DVD в терминале, набрав команду

```
cp /dev/dvd somemovie.iso
```

Их воспроизводит большинство видеопроигрывателей. Особенно хорошо подходит *VLC*, так как предоставляет все меню и спецвозможности.

Если DVD зашифрован, придется установить *libdvdcss* на всех устройствах, предназначенных для воспроизведения файла. Альтернатива – создать незашифрованную копию с помощью *vobcopy* или *dvdcpy* (часть пакета *OGMRip*):

```
dvdcpy /dev/dvd
```

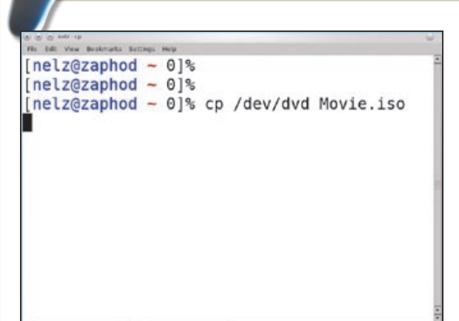
Однако такие файлы не умеет воспроизводить *MythVideo*.

Если у Вас мало памяти, обдумайте сжатие файлов. Качество снизится, но DVD кодируются с достаточно высоким битрейтом, выдерживающим значительное сжатие без особой потери качества. Одна из программ, пригодных для этого – *OGMRip* (http://ogmrip.sourceforge.net), она допускает захват видео в различных форматах, а не только в AVI, как другие. Это важно, потому что форматы типа Matroska хранят информацию о частях DVD.

Откройте *OGMRip* и выберите DVD. Появится список частей в первом фильме. При проверке настроек выберите для кодирования одну часть: это ускорит дело. Перед кодированием DVD надо создать профиль, определяющий вывод. Если вы кодируете для проигрывания на конкретном устройстве, то сайт *OGMRip* предлагает ряд готовых профилей; их можно скачать и импортировать. Иначе – выберите Edit > Profiles, нажмите кнопку New и дайте новому профилю имя, а затем нажмите кнопку Edit, чтобы задать параметры. Ваша забота – только формат и размер; на вкладке General можно задать фиксированный битрейт, либо указать размер и позволить *OGMRip* вычислить битрейт, установить кодек и качество на вкладке video. Для прочих параметров сойдут значения по умолчанию. Затем закройте окно профилей, выберите часть и нажмите кнопку Extract, затем выберите только что созданный профиль – и идите ставить чайник, потому что эта операция требует времени.

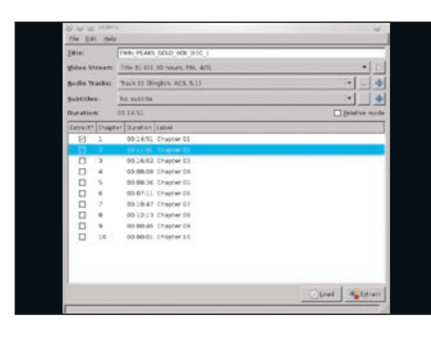
Если преобразуются файлы, а не DVD – или Вы уже скопировали DVD в ISO-файлы – можно поставить операцию кодирования в очередь, а затем настроить еще одну. Когда вся очередь будет готова, запустите *OGMRip* на кодирование всех файлов одновременно. **ГМ**

Шаг за шагом: Копирование DVD на жесткий диск



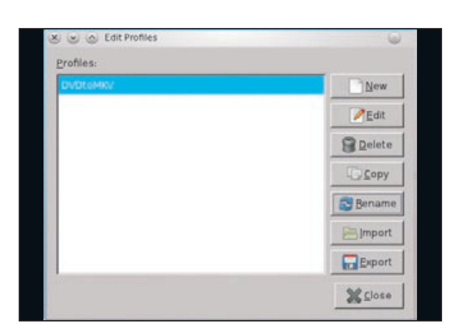
1 Копирование диска

Простейший способ копирования DVD с наивысшим качеством – списать его целиком командой **cp**, но полученный файл будет большого объема.



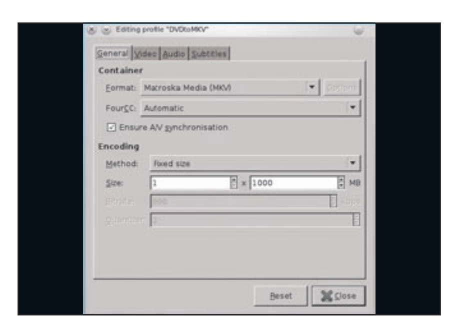
2 Выбор части

OGMRip сохраняет части DVD, и можно кодировать по части за раз, чтобы быстрее проверять настройки.



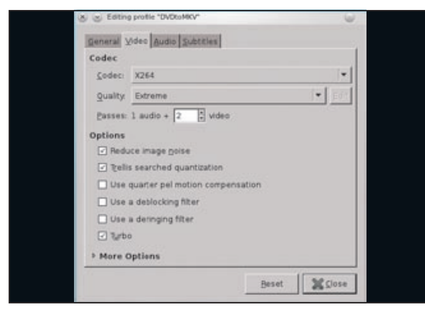
3 Создание профиля

Профилей по умолчанию нет, поэтому либо создайте пару-тройку, либо скачайте и импортируйте несколько профилей с сайта *OGMRip*.



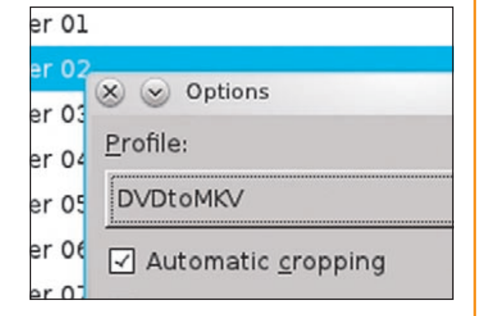
4 Выбор формата

Для максимальной совместимости выберите AVI, или, для получения более широких возможностей, один из более современных форматов. Кроме того, здесь можно указать размер файла или битритейт.



5 Выбор кодека

Настройки кодека и качества существенно влияют на качество итогового видео и на время, необходимое для его получения.



6 Запуск процесса

Нажатие кнопки Extract дает еще несколько вариантов выбора, включая установку очередности преобразований перед началом кодирования.

Лучшее в мире новое ПО с открытым кодом

LXFHotPicks



Ник Вейч
Компилируя HotPicks, Ник частенько употребляет *Curses*. Пошлите на известный адрес сообщение о своем любимом языке программирования.

KiCad » DMIDecode » Bandwidth » Smartie-utils » Wings3D » Figlet » Din » Kernel Panic » PokerTH » Powertop » Hotot

Разработка электроники

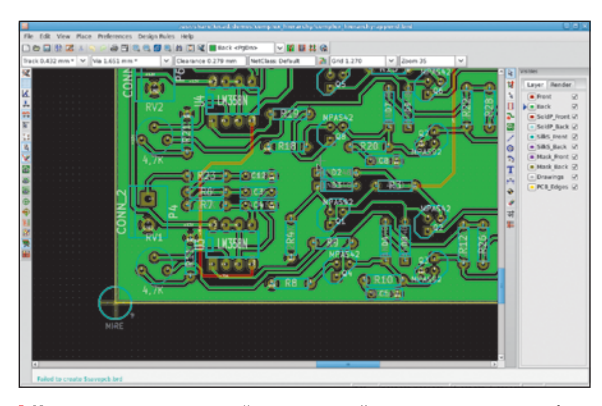
KiCad

Версия 20110123 **Сайт** https://launchpad.net/kicad

Революция Arduino дала старт новой волне интереса к электронике, что, безусловно, дело хорошее – особенно, как оказалось, для линуксоидов: появилась куча программ для электронных устройств, готовых помочь в любых начинаниях, от конструирования электросхемы до создания итоговой раскладки печатной платы. Печать 3D-схем прямо с ПК – это ведь только вопрос времени, верно?

Программ полным-полно, но безусловно стоит внимания проект *KiCad*. Как видно из названия, это приложение GUI для KDE, которое в данном случае перекрывает целый инструментарий. В главном окне запуска легко переключаться между проектами и потом погружаться в детали – будь то создание проекта с нуля, отбор элементов схемы, разработка печатных плат или редактирование файлов Gerber. Не хватает только хорошего эмулятора схемы, но можно экспортировать Spice netlists, более или менее стандартные для этой цели. Благодаря достойному набору библиотечных моделей с разнообразными пакетами компонентов, вы также сможете генерировать 3D-изображение своей готовой схемы, которое не просто красиво выглядит, но также поможет вам увидеть и решить физические проблемы с дизайном.

> «В главном окне легко переключаться между проектами.»



› Инструмент схем не такой симпатичный, как некоторые, но работу он выполняет.

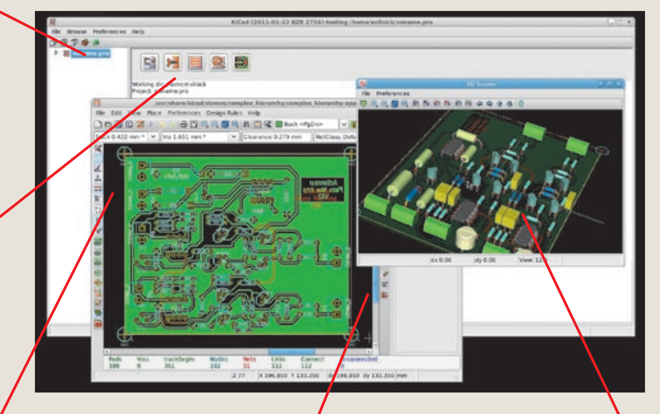
Завершив разработку своей схемы, вы можете экспортировать ее в массу форматов, соответствующих различным типам производства, так что будет весьма несложно ее изготовить через онлайн-сервис или распечатать шаблоны протравки для самостоятельного изготовления, если ваши руки достаточно умелые.

В прошлом году разработка *KiCad* перешла на Launchpad, и немедленным эффектом этого стала доступность версий разработчиков. Сначала следует установить самый свежий стабильный релиз, поскольку он вытащит все зависимости, о которых вы даже не думали, и, что крайне важно, установит библиотеки компонентов, которых нет в исходных репозиториях. Затем, если у вас в системе установлен *Bzr*, проверьте наличие самого свежего кода командой

```
bzr branch lp:kicad kicad.bzr
```

Конечно, это код в разработке, так что кому как повезет. *KiCad* имеется в репозиториях практически всех основных дистрибутивов, и оттуда всегда можно взять версию постарше.

Исследуем KiCad



Панель проектов
Текущие проекты отображаются в левой панели – щелкните здесь, чтобы увидеть наличные компоненты.

Запуск действий (активности)
Запустите определенное действие, щелкнув по одному из значков инструментов.

Скроен по фигуре
Хотя разные части этого пакета выглядят похоже, в них всех имеются разные инструменты для конкретных работ.

Основные инструменты
PCB Layout – только один из здешних важных инструментов. Он довольно прост в использовании, если к нему привыкнуть.

Обретение формы
Вы даже можете получить аккуратное 3D-отображение вашей переполненной рабочей области, что может оказаться очень удобным при проверках зазоров.

Системный инструмент

DMIDecode

Версия 2.11 **Сайт** www.nongnu.org/dmidecode

На заре Linux новая установка требовала знания огромного количества информации. Вы бы ответили на кучу вопросов установщика, пока не выдалось бы нечто вроде «Пожалуйста, выберите чип UART для вашего серийного интерфейса». Тут полагалось ругнуться, приостановить процесс и взяться за отвертку, потому что, честно говоря, не было иного способа получить эту информацию, кроме как посмотреть этот номер прямо на чипе – ну, то есть, если вы вообще могли его найти.

С тех пор программы установки слегка шагнули вперед, и вполне можно установить и успешно запустить Linux, даже не зная – и не заботясь – о спецификации вашего «железа». Ура! Но все же для тех случаев, когда это важно, нашелся ли менее деструктивный способ познания?

Система, которая обладает основной информацией о вашем оборудовании – отнюдь не ваша установка Linux. Да, ядро неплохо находит нужные данные, но система, знающая все обо всем, проживает на материнской плате: это BIOS. К счастью, поскольку BIOS должен общаться с ОС, на это имеются протоколы и стандарты, и таким-то образом мы и можем до капли извлечь всю информацию о работающей системе.

К сожалению, *DMIDecode* вряд ли перечислит вам названия всех чипов, соединенных с оборудованием вашего компьютера, зато многое скажет о ключевых элементах: например, какие слоты памяти заполнены, насколько велик ваш кэш L2 и кто изготовил вашу материнскую плату.

> «Он скажет вам, какие слоты памяти заполнены и велик ли кэш L2.»

```
File  Edit  View  Bookmarks  Settings  Help
SMBIOS 2.4 present.
38 structures occupying 1190 bytes.
Table at 0x000F0100.

Handle 0x0000, DMI type 0, 24 bytes
BIOS Information
        Vendor: Award Software International, Inc.
        Version: F11
        Release Date: 08/20/2010
        Address: 0xE0000
        Runtime Size: 128 kB
        ROM Size: 8192 kB
        Characteristics:
                PCI is supported
                PNP is supported
                BIOS is upgradeable
                BIOS shadowing is allowed
                Boot from CD is supported
                Selectable boot is supported
```

› *DMIDecode* расскажет о вашей системе все, что нужно, и корпус вскрывать вам не придется.

Он также может – и это весьма полезно – определить текущие настройки BIOS, которые очень пригодятся при использовании диагностических приложений.

DMIDecode достаточно просто компилируется из имеющегося исходника (есть Makefile), так что нет нужды долго рассказывать о поиске пакетов и т. п. Однако запускать его следует с правами root, потому что требуется доступ к системным устройствам для общения с BIOS.

Тестирование производительности

Bandwidth

Версия 0.25a **Сайт** http://caladan.tk

Пытаясь добиться от своей системы лучшей отзывчивости и большей эффективности, вы, весьма вероятно, в какой-то момент поддались соблазну прикупить сдвоенное (или даже утроенное, в зависимости от вашего пыла) ОЗУ – самое свежее и быстрое на вид. Возможно, вы даже завели всякие штучки с лампочками, которые периодически вам подмигивают, чтобы вы осознали, как дорого они стоят. Вы обольщаетесь, что теперь все намного ускорилось и окна открываются как минимум на 30 % быстрее, чем раньше, но нельзя ли это измерить на самом деле?

Систем тестирования производительности [benchmarking] памяти немало, но этот инструментик стоит рассмотреть. Он старается построить полный профиль быстродействия памяти на основе циклических тестов с данными разных размеров в последовательных и случайных тестах чтения / записи. Он также по возможности задействует спецфункции реестра, и может отдельно протестировать кэши L1 и L2, позволяя в полной мере осознать важность их роли в производительности ОЗУ вашего компьютера.

Поскольку код компилируется для 32- или 64-битных систем, вы даже заметите разницу при установке 64-битной версии вашего привычного дистрибутива. *Bandwidth* также компилируется для чипов ARM, и для выяснения, быстро ли работает ваше встроенное устройство или телефон с Android, сгодится тот же код. Большинство полученных автором результатов на реальных системах, похоже, подтверждают расхожие мнения о том, как память и разные виды доступа к ней влияют на производительность – например, что у 64-битных приложений солидное преимущество по скорости и что большой кэш L1 лучше, чем гораздо больший кэш L2.

> «Для выяснения, быстр ли телефон на Android, сгодится тот же код.»

```
File  Edit  View  Bookmarks  Settings  Help
This is bandwidth version 0.25a.
Copyright (C) 2005-2010 by Zack T Smith.

This software is covered by the GNU Public License.
It is provided AS-IS, use at your own risk.
See the file COPYING for more information.

Using 128-bit and 64-bit data transfers.
Notation: kB = 1024 B, MB = 1048576 B.

CPU speed is 2661 MHz.

Sequential read (128-bit), size = 256 B, loops = 977534976
Sequential read (128-bit), size = 512 B, loops = 315097088
Sequential read (128-bit), size = 768 B, loops = 326280654
Sequential read (128-bit), size = 1 kB, loops = 243597312,
Sequential read (128-bit), size = 2 kB, loops = 121896960,
Sequential read (128-bit), size = 3 kB, loops = 81569230,
                                        loops = 61046784,
```

› Собрав данные и составив элегантный график, можно сравнить его с теми, что прилагаются к программам.

Bandwidth недавно ввел также и функцию мониторинга ширины полосы сети, и это прекрасно – хотя настройка всех необходимых узлов довольно сложна. Но если вы действительно хотите протестировать, на что способна ваша система, этот удобный инструмент обработает по крайней мере два пункта из вашего перечня позиций.

Эзотерические инструменты

Smartie-utils

Версия 1.0 **Сайт** http://code.google.com/p/smartie-utils

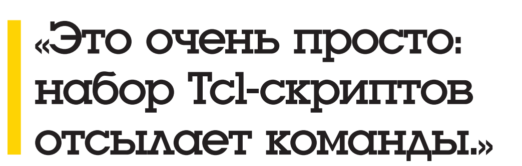


› **Дивитесь! Из всех рассмотренных здесь проектов это первый, у которого есть рисунок, а не экранный снимок!**

Давным-давно, в далекой, далекой галактике… Ох нет, лет пять назад в доме напротив. Ну, короче, в не столь отдаленной точке пространственно-временного континуума – но достаточно далекой, чтобы среднестатистическая личность сочла ее прошлым – самая крутейшая штуковина, которую вы могли воткнуть в свою Linux-машину через Blu-Tack, был LCD-дисплей. Эти устройства, всех цветов радуги, подсоединялись к внутреннму разъему последовательного порта и программировались для вывода такой полезной информации, как текущая загрузка CPU, температура или время, оставшееся до завершения свежезапущенной компиляции ядра.

Такая штука на передней панели корпуса стоила фунт-другой, но в наши дни уже странно, что они не распространяются вмонтированными в *Linux Format*. Одно из наиболее привлекательных по цене из этих устройств, известное как Smartie, производит в Китае фирма Sure Electronics. Однако, при всех хитростях дизайна от Sure, в нем нет главного, потому что все прилагаемые драйвера имеют закрытый код, причем для Windows (хотя, справедливости ради, отметим, что они более или менее отвечают фактическим стандартам).

Для пользователей Linux существует отличный *LCDProc*, но его серверно-клиентская ориентированность и необходимые файлы настройки сильно осложняют задачу, если вам всего-то и нужно, что походя отображать проклятия по адресу Майка. *Smartie-utils* идут немного иным путем, предлагая простой способ отображения текста на дисплее. Это действительно очень просто: набор Tcl-скриптов отсылает команды устройству и направляет на него текст. Возможно, он делает не все на свете, но для простых сообщений или установки на каком-либо встроенном устройстве в нем есть все необходимое.



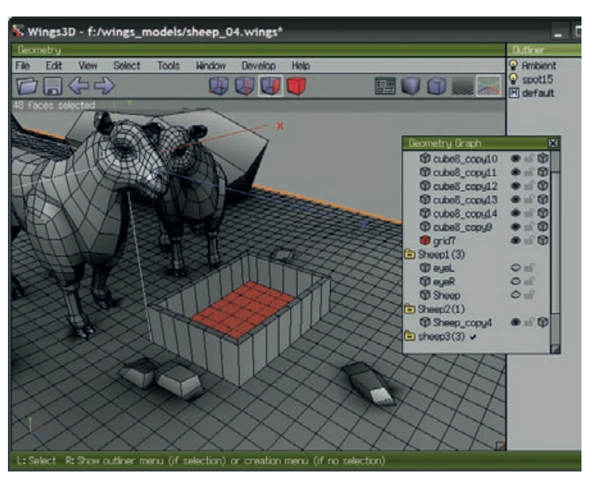
> «Это очень просто: набор Tcl-скриптов отсылает команды.»

Это первый релиз кода, и, возможно, при наличии обратной связи и поддержки автор внесет в него функции, которых, как вам кажется, там не хватает.

Программа 3D-моделирования

Wings3D

Версия 1.4 **Сайт** www.wings3d.com



› **Как мы ни старались, ничто из созданного нами не превзошло этой чудной картинки с сайта, где овцы пьют… хм… кровь?!**

Наш журнал, пожалуй, не тянет на сообщение новостей о новой версии Wings, потому что способен сделать это всего лишь в двух измерениях. Однако, если вдуматься, программа все равно создает 3D-модели на 2D-носителе – вашем мониторе. Пускает пыль в глаза.

Как обычно бывает с подобными приложениями, вы начинаете с создания простого «примитива», потом обрабатываете его и нарезаете, пока не получите должный результат. Подобных инструментов в Linux уже немало, но *Wings3D* обладает рядом достойных функций, и хотя интерфейс выглядит никак не линуксово, ему, возможно, удастся чем-нибудь вас соблазнить.

Помимо стандартных инструментов придания формы объектам, здесь имеется очень удобное магнитное устройство – работает оно по принципу магнитов, применяемых для ремонта кузова автомобиля, и используется для изменения формы вершин; а автоматическая развертка заодно упрощает процесс текстурирования. Программа моделирования также поддерживает массу форматов импорта и экспорта, и если вы оттачиваете объект 3DLS, LightWave или Wavefront (онлайн таких много) или жаждете экспортировать в формат типа VRML или POV-Ray, будьте покойны.

Для поддержания своего многомерного мира *Wings* использует любопытный и, возможно, недооцененный язык Erlang. Взяв *Wings* из своего стандартного репозитория, вы, возможно, получите более старую версию, но зато еще и сам язык и все необходимые дополнительные библиотеки.

> «Программа моделирования поддерживает импорт и экспорт.»

Если же вы выберете путь самостоятельной компиляции – прежде чем жать на **make**, установите пакет Erlangesdl и настройте переменную **ESDL_PATH**. Это можно сделать, запустив нечто вроде

```
export ESDL_PATH='/usr/lib64/erlang/lib/esdl-1.0.1/'
make
```

А еще можно загрузить Deb-пакеты с домашней страницы.

Инструмент для заголовков

Figlet

Версия 2.2.3 **Сайт** www.figlet.org

Бывают такие приложения с открытым кодом, которые преобразуют саму действительность, а бывают небольшие умные приложения, делающие нечто полезное и необычное. Вам решать, к какой категории относится *Figlet*; авось наше краткое описание вам поможет.

Figlet отчасти похож на программы для заголовков, используемые для печати текста крупного размера в начале каждого задания на строчном принтере, разделяемом всем отделом. Заголовки облегчают рабский труд по обнаружению своей распечатки, но это не главный элемент их крутизны. Гигантские символы создавались повторным обращением к стандартному набору символов ASCII и совмещением соответствующих символов для получения супер-читаемых букв. И то же самое делает *Figlet*.

Шрифты его сами по себе являются хитроумными описаниями того, как создавать отдельные буквы, и на отработку каждого FLF-файла нужно огромное мастерство и терпение. На сайте имеется раздел, целиком создаваемый пользователями, где можно просмотреть и скачать гарнитуры шрифтов.

Теперь, когда в отделах уже нет строчных принтеров, чаще всего подобные файлы заголовков используют для подписей на онлайн-форумах – тех, что занимают место и не факт, что радуют глаз. Но у них есть законное применение в затейливых слоганах баннеров, и если вам где-нибудь на помойке попадется IBM 1403, вы уж будете знать, что делать.

Figlet – простая программа для терминала, и для ее компиляции ничего этакого не требуется. После распаковки исходника, возможно, неплохо будет отредактировать Makefile или настроить переменные **$ENV**, перечисленные в readme, чтобы обеспечить установку куда надо и чтобы все заработало. Однако устанавливать ее вовсе не обязательно, потому что командная строка дает возможность указать путь шрифта.

› Может, практического применения у нее и нет, но зато ваши лозунги будут видны из космоса.

> «Если вам попадется IBM 1403, вы уж будете знать, что делать.»

Аудиоинструмент

Din

Версия 1.4.1 **Сайт** http://dinisnoise.org

Если вы не доверяете рекурсивному акрониму этого приложения, Din Is Noise [Din Is Noise значит «Грохот – это шум», – *прим. пер.*], обратитесь к моим соседям, которые явно не в восторге от моих текущих подвигов в сфере аудиомастеринга. Тем не менее, одно из удивительных и классных свойств *Din* – то, что возиться с ним – одно удовольствие.

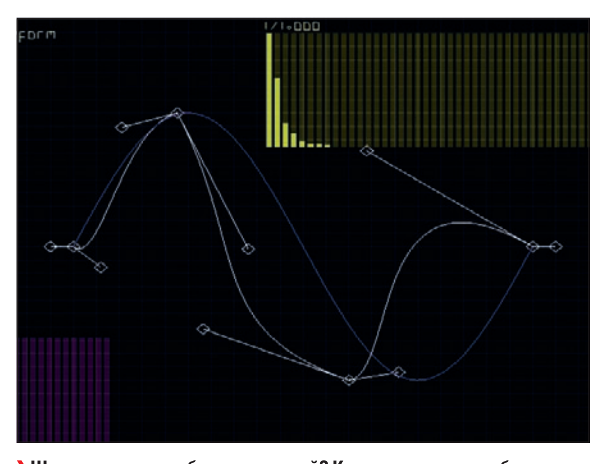
Раз можно рисовать с помощью света, почему бы не рисовать с помощью музыки? Две эти области, в общем безразличные большинству хакеров, пересекаются в этом генераторе волновых форм плюс синтезаторе, где вам на откуп отданы повторяющиеся волновые формы. Запустите его – он сразу же ругнется – и нажмите на F1, чтобы получить подсказку про работу в нем, хотя вам и самим по силам разобраться в происходящем, просто переключаясь по разным экранам редактирования.

Лучше всего, наверное, начать с редактора волновых форм, работающего с кривыми Безье для управления амплитудами. Используйте клавиши с цифрами для переключения на отображение разных редакторов, и функциональные клавиши, чтобы менять параметры отображения на экране.

Вы можете варьировать масштаб экрана и делать прокрутку клавишами W, A, S и D, часто применяемыми в играх FPS. На сайте имеются достаточно подробные объяснения по редакторам и их разнообразным настройкам, так что вы, возможно, захотите и их тоже скачать – это намного проще, чем читать инструкции с экрана в векторных шрифтах.

Это поможет, если вы хоть чуть-чуть знакомы с технической стороной производства аудио, но только если вы точно представляете себе, какой звук намерены создать. Самая сильная сторона *Din* в том, что это отличное орудие эксперимента по отпугиванию диких животных. А еще в него встроен рудиментарный IRC-клиент.

Для компиляции *Din* понадобятся Jack, Tcl и, хотя это отчасти и раздражает, библиотеки *libircclient* (из-за встроенного клиента для чата). Последняя не всегда входит в пакеты, но компиляция из исходника не вызывает сложностей.



› Шум какого типа обозлит соседей? Как выясняется, любого, если вы у руля *Din*.

> «Сильная сторона Din в том, что это отличное орудие эксперимента.»

HotGames Развлекательные приложения

Стратегия реального времени

Kernel Panic

Версия 4.1 **Сайт** http://springrts.com/wiki/Kernel_Panic

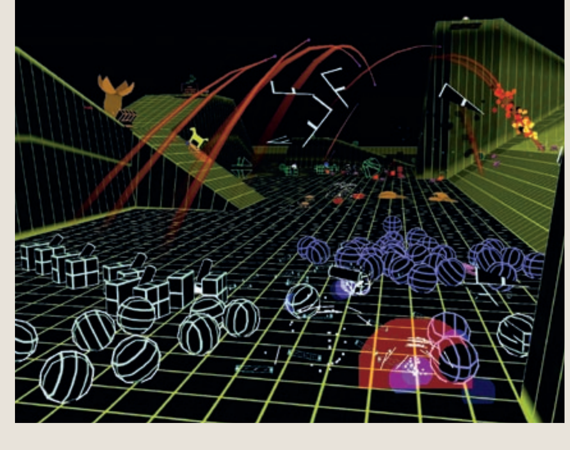
Проблема всех этих стратегий реального времени – таких, как *Age of Empires*, *Starcraft* и прочих – ужасные траты времени на сбор веточек, камешков и золотых слитков для постройки того, что вы там затеяли строить. Сбор ресурсов, если быть честными, всего лишь отвлекает вас от главной интриги: группировки большого числа боевых единиц и полного разгрома противника.

Tron-подобный *Kernel Panic*, размещенный в микромире электронных схем вашего компьютера, по сути, не имеет ресурсов, которые надо собирать. Здесь есть цифровые отдушины, которые выделяют двоичную энергию, ожидающую превращения в различные постройки. Эти юниты могут затем колонизировать другие «фабрики», чтобы создать еще больше юнитов, причем каждый завод производит юниты до тех пор, пока принадлежит вам. Да, но вот незадача: ваши процессы – не одни воины в поле. Скрытые процессы AI тоже работают, претендуя на цифровые кузни как на свою собственность, и выживает только один.

При игре применяется другая стратегия: управление множеством юнитов в кажущейся бесконечной бойне. Графика в игре векторноподобная, отлично анимирована – вплоть до взрывов, и добавляет острых ощущений при виде избиения вражеских отрядов.

Kernel Panic – это набор модулей для движка *Spring*, который также нужно установить. Чтобы развернуть ее, просто скопируйте режимы и карты в директорию**.spring** в вашей папке **home**, затем запустите *Springlobby* для завершения настройки. Если игра покажется вам чересчур безумной, есть хороший набор пакетов модулей для настройки игрового процесса в основном интерфейсе *Spring*.

> «При игре — другая стратегия: управление множеством юнитов.»



› **О нет! Они убили 0100101! Он был лучшим, поймите. Самым лучшим.**

Играя в одиночном режиме, позаботьтесь о выборе искусственного интеллекта *Kernel Panic*, потому что обычный искусственный интеллект должным образом не работает. Да, в текущей версии пока еще присутствуют недоработки, но она уже сейчас фантастически играбельна и стоит того, чтобы потратить на нее полчасика.

Карточная игра

PokerTH

Версия 0.8.3 **Сайт** http://www.pokerth.net

Рик Блейн [Rick Blaine] из фильма «Касабланка» любил попытать счастья в рулетке, и все мы знаем, что 007 желал набрать чистую девятку в баккара [карточная игра, – *прим. пер.*], но что заманит в казино Викторию Корен [Victoria Coren – знаменитая победительница покерного турнира EPT в Лондоне, – *прим. пер.*]? Дык покер!

Игра возродилась с тех пор, как международные азартные игры стали показывать по телевизору – в частности, техасский вариант Hold'em. Сразу же проясним: мы не предлагаем вам отыграть свою оплату подписки на *Linux Format*. Хотя, если вы не играете на деньги, вам не повредит небольшой блеф или контрблеф.

Существует море карточных игр – включая разновидности покера – доступных для Linux, но *PokerTH* стоит особняком по ряду причин. Во-первых, он отлично сделан: графика очень приятная, и очень легко понять, что происходит. Здесь есть даже небольшой экран помощи, который рассчитает текущие возможности получения самых разнообразных выигрышных комбинаций.

Вторая причина – в наличии сетевой поддержки, позволяющей участвовать в онлайн-матчах. Сайт отслеживает ваши результаты и рассчитывает всемирный рейтинг. Похоже, *PokerTH* весьма популярна: в любое время идет порядка сотни игр онлайн, и вы не будете обделены оппонентами. Встроенные AI-игроки тоже неплохи – у них есть определенные умения, и они не сторонятся стратегии блефа.

> «В любое время идет порядка сотни игр онлайн.»



› **Если мне позволено будет сказать, вы, как я вижу, остались без тузов…**

PokerTH сейчас получил прописку всего в нескольких репозиториях, включая Ubuntu, но компилировать его вовсе не сложно. Даже не нужно его устанавливать – просто загрузите статично скомпилированную версию и запустите ее прямо из директории.

Монитор энергопотребления

Powertop

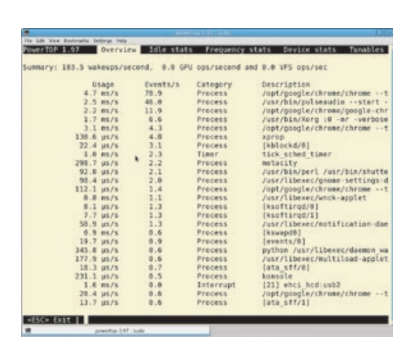
Версия 1.97 **Сайт** http://kernel.org/pub/linux/status/powertop

Из всех вариантов *top*, *Powertop* всегда ходил у нас в любимчиках. Конечно, другие утилиты тоже дают полезную и точную информацию о работающих процессах, использовании памяти и активности диска, но Powertop умеет оценить то, что, по нашему мнению, вы бы ни за что адекватно не измерили – а именно, энергопотребление.

Среднему компьютеру незачем особо задумываться, сколько энергии он потребляет за данный промежуток времени, но бывают ситуации, когда это знание весьма важно. Например, в крупных центрах обработки данных самая большая проблема – нагрев. Тепло – побочный продукт переработки энергии компьютером, чтобы гонять электроны, и все, что служит экономии энергии – дело хорошее.

Powertop изначально разрабатывался Intel с целью отслеживания потребления энергии приложениями, но вполне применим для выяснения, какие именно части вашего Linux слишком налегают на харчи.

Эта новая версия, предвестник релиза 2.0, здорово отладила применяемые алгоритмы. Однако основная идея в том, что процессы, которые заставляют CPU работать в режиме повышенного энергопотребления – это плохие процессы, если только они не являются продуктивными. Обычно это вышедшие из-под контроля демоны или ненужные сервисы. Программа уже внедрялась в призвание к порядку ядра и приложений вроде *Firefox*, чтобы те вели себя прилично.



› В зависимости от способностей вашей системы, клавиши со стрелками делают переход с одной таблицы системных данных на другую.

Клиент Twitter

Hotot

Версия 0.9.5 **Сайт** http://hotot.org

Судя по всему, чтобы попасть на вечеринку Twitter, нужно дурацкое название. У KDE имеется *Chokoq*, у Gnome – *Gwibber*, а фанаты кросс-платформенности теперь получили *Hotot*.

Hotot позиционируется как легковесный клиент, но обладает массой функций. Можно загружать мультимедиа на три самых популярных хостинговых сайта, группировать твиты и ретвиты в протоколируемый диалог и добавлять в базовое приложения расширения. Сливки этих приложений – геолокация с помощью Google Maps, поддержка Instapaper для чтения документов оффлайн и переводчик – и все это закатано в ровный, гладкий и современный GUI.

Благодаря чудесному отображению протоколируемого диалога, он мог бы стать победителем в нашей номинации «Самый симпатичный клиент Linux Twitter» (если бы такая существовала), а переход с одного отображения на другое подобен переходу в устройстве iOS. Однако симпатичные переходы имеют свою цену, и *Hotot* лишен отзывчивости его родичей в KDE и Gnome. Экран обновляется медленно, и появления окон требуется подождать. Это похоже на web-интерфейс с хорошим рендерингом, например, и данной проблемой он, судя по всему, обязан своим корням, уходящим в Python. Но это мелкая критика по адресу пока еще альфа-релиза, и нам легко представить, что через год мы будем рекомендовать *Hotot* как стоящий вне конкуренции, особенно для планшетных устройств.



› Развлекайте легионы своих друзей по Twitter, злостно эксплуатируя PulseAudio.

Также вышли

Новые и обновленные программы, тоже достойные внимания...

» The Legend Of Edgar 0.75
Обновление этого фаворита HotPicks добавляет новые области для исследования.
http://www.parallelrealities.co.uk/projects/edgar.php



› Позвольте одолжить ваш сарай? Обещаю не трогать крыс...

» Samba 3.4.10
Свежие устранения ошибок в самой жизнеспособной программе разделения доступа к файлам.
http://www.samba.org

» MikeOS 4.1.1
Когда Linux погрязнет в судах из-за авторских прав, мы все будем ею пользоваться. Готовьтесь!
http://mikeos.berlios.de/

» RemoteBox 0.6
Управляйте своими виртуальными машинами с разных компьютеров через этот аккуратный клиент.
http://remotebox.knobgoblin.org.uk

» Pepper 0.1.4
Оторвитесь от кодирования и постройте симпатичные графики своего прогресса!
http://scm-pepper.sourceforge.net

» TwoLAME 0.3.13
Исправленные ошибки и первое за два года обновление программы декодирования MPEG-аудио.
http://www.twolame.org

» Feh 1.11
Быстрая, легкая программа для просмотра изображений, концентрирующая внимание на высокопроизводительной загрузке и масштабировании.
http://feh.finalrewind.org/

» DirSyncPro 1.4
Программа на Java для настольного ПК, которая синхронизирует файлы, находящиеся в разных местах.
http://www.dirsyncpro.org

» Fx Movie Manager 4.8
Интерфейс для работы с видеофайлами.
http://stagione.interfree.it

На диске

Новые среды рабочего стола и перерождение OpenOffice.org...



Прочтите всю информацию, необходимую для использования DVD! Если вы новичок в Linux, откройте файл index.html на диске и перейдите в раздел Справка, где вы найдете руководства по вопросам, перечисленным справа:

Майк Сондерс, редактор диска
mike.saunders@futurenet.com

» Что такое Linux?
» Что такое дистрибутив?
» Загрузка ПК с DVD
» Разбиение жесткого диска на разделы
» Навигация по файловой системе
» Использование командной строки
» Установка программ

Среда рабочего стола

KDE 4.6

KDE 4 – одна из самых противоречивых программ за всю историю Linux. Одни пользователи полагают, что она обеспечила невероятный прогресс рабочего стола; согласно другим, это была ошибка и разработчикам следовало развивать кодовую базу 3.x. Так или иначе, нельзя отрицать, что первым релизам 4.x не хватало отделки и многих функций, но в 4.6 внесено множество поправок. См. наш материал в разделе «Обзоры».

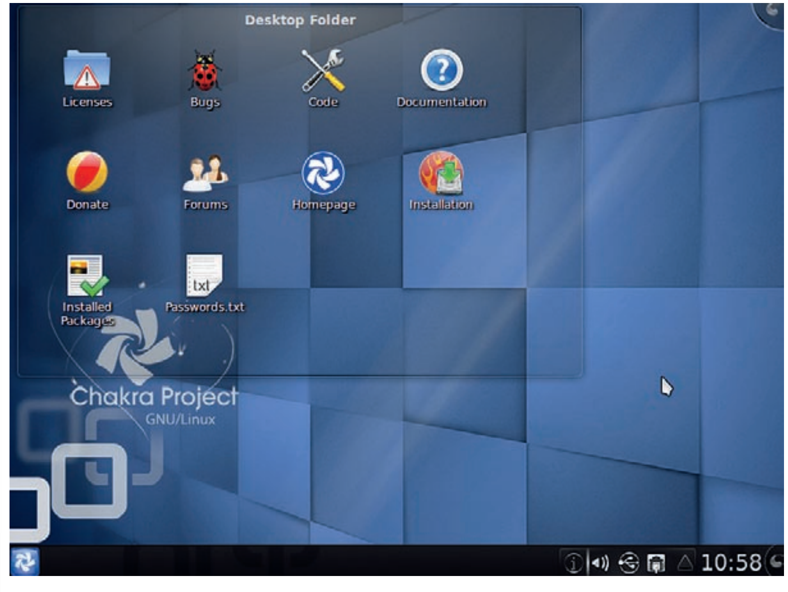
Есть два способа познакомиться с KDE 4.6 с **LXFDVD**: простой и только для хакеров. В первом случае можно просто загрузить Live CD с дистрибутивом Chakra, еще не попадавшим на DVD: сайт проекта – www.chakra-project.org. ISO-образ Live CD находится на **LXFDVD** (файл **chakra-2011.02-kde460-i686.iso** в каталоге **/Distros/KDE_Live/**). Запишите его на CD-R либо подключите к виртуальной машине VirtualBox или любого другого гипервизора.

По словам создателей этого дистрибутива, их целью было предоставить «быстрый, дружественный к пользователю и исключительно мощный live CD», чего мы уже наслушались в мире дистрибутивов; однако сильная сторона Chakra – очень быстрый темп разработки. KDE 4.6 вышел всего за несколько дней до окончания работы над DVD, и команда Chakra умудрилась очень быстро втиснуть туда live-дистрибутив с новым рабочим столом.

> «Настройки Chakra просты, и вам дадут первозданный KDE.»

Настройки KDE в Chakra очень просты, и вам дадут первозданный KDE 4.6 от разработчиков, не усложненный узкоспецифическими для дистрибутива заплатами и настройками.

Вторая опция для знакомства с KDE 4.6 – исходный код, представленный в разделе Desktop на DVD. Компиляция этих tar-архивов – сложная задача, рекомендуемая только в том случае, если вы разбираетесь в командной строке и компиляции программ из исходника. Полное руководство по компиляции – на http://techbase.kde.org/Getting_Started/Build/KDE4.



› **Chakra — один из первых дистрибутивов, включивших рабочий стол KDE 4.6, и загрузить его с LiveCD невероятно просто.**

Важно ВНИМАНИЕ!

Перед тем, как вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим.

Диски *Linux Format* DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция *Linux Format* не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензии.

Бракованные диски

В маловероятном случае обнаружения бракованного диска *Linux Format*, просим связаться с нашей группой поддержки по адресу disks@linuxformat.ru для получения содействия.

Офисный пакет

LibreOffice 3.3

Ветвление в мире свободного ПО – часто процесс болезненный, но, как мы видели по *XFree86* и *X.org*, необходимый, если проект стал пробуксовывать. Так было и с *OpenOffice.org* – разработчики поняли, что разработка новых функций идет недостаточно быстро (отсюда и ветвь *Go-oo*). Недавнее приобретение Oracle Sun и прекращение работы над OpenSolaris тоже наводили на сомнения.

Итак, группа кодеров начала работу над проектом *LibreOffice*, задавшись целью освободить код *OOo* от корпоративного влияния и ускорить процесс выхода новых поправок. Пока что успех был внушительным: все основные дистрибутивы планируют перейти на него, и, благодаря реформам *Go-oo*, он уже загружается чуть быстрее. К сожалению, команда ничего не предприняла, чтобы процесс установки стал более продуманным.

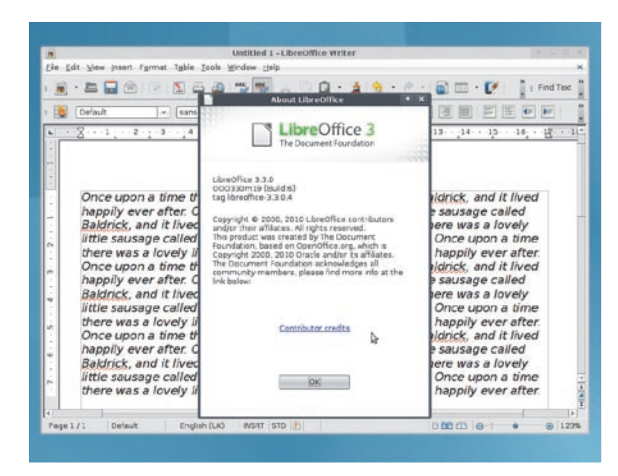
В разделе Desktop/LibreOffice на DVD вы найдете два файла: один с DEB **в имени** файла и один –с RPM. Скопируйте соответствующий файл в свою директорию **home** — DEB для дистрибутивов на базе Debian; RPM **для Fedora, Mandriva и Open**SUSE. Пользователи Deb должны открыть терминал и ввести такие команды (используйте автозаполнение, чтобы не вводить имена файлов полностью):

```
tar xfv LibO_3.3.0_Linux_x86_installdeb_en-US.tar.gz
cd LibO_3.3.0rc4_Linux_x86_installdeb_en-US/DEBS/
```

Потом переключитесь на root (**sudo bash** или **su**, в зависимости от вашего дистрибутива) и введите **dpkg -i *.deb** для установки.

Пользователи RPM должны запустить

```
tar xfv LibO_3.3.0_Linux_x86_installrpm_en-US.tar.gz
cd LibO_3.3.0rc4_Linux_x86_installdeb_en-US/RPMS/
su
rpm -ivh *.rpm
```



› *LibreOffice* по виду, вкусу и запаху подобен *OpenOffice.org*.

Каким бы путем вы ни пошли, теперь можно удалить файл **.tar.gz** и директорию, созданную при распаковке, и запустить пакет командой

```
/opt/libreoffice/program/soffice
```

Более подробную информацию о пакете вы найдете в соответствующей статье раздела «Обзоры». А еще – пройдите по адресу http://lists.freedesktop.org/mailman/listinfo/libreoffice и отправьте сообщение с жалобой на дурацкую процедуру установки.

Другие программы

Xfce 4.8, инструменты разработки, подкасты...

KDE – не единственный рабочий стол, отмечающий в этом месяце выход нового релиза. *Xfce*, легковесный рабочий стол на основе *GTK*, который многие считают щадящей память альтернативой Gnome, дорос до версии 4.8. Особых поводов ликовать тут нет – теперь файловый менеджер умеет просматривать ресурсы с распределенным доступом, а в остальном изменения незначительные. *Xfce* идет в виде исходного кода в разделе Desktop диска; в **BUILD.TXT** имеется руководство.

В разделе Development у нас *GTK 2.24.0*, новый релиз популярного инструментария для Gnome. Это небольшой шаг на пути к *GTK 3*, так что многие функции исключены, и добавлены новые API. Загляните в **NEWS** внутри tar-архива: там приведен подробный список изменений. В этом разделе вы также найдете *KDevelop 4.2*, обновление популярной интегрированной среды разработки. Подробности – в разделе «Учебники».

Мы здесь, в Башнях **LXF**, тоже не лыком шиты – нам, например, хорошо известно, как сделать чертовски классное интернет-радиошоу. Каждый две недели мы создаем TuxRadar Podcast: собираемся вокруг микрофонов и пережевываем главные темы дня, касающиеся Linux. Также мы делимся своими открытиями в области программ, заставляем некоторых отстаивать свой выбор программ в нашем разделе «На скамье подсудимых» и делаем множество других интересных вещей. Самые свежие эпизоды вы обнаружите в разделе Magazine/Podcast на DVD.

› *Xfce 4.8* теперь позволяет добавлять папки как пункты в панели, делая процесс просмотра файлов супербыстрым.

И это еще не все!

Пять классных игр...

После всех этих серьезностей, почему бы не поразвлечься с помощью нашего раздела Games? Главный хит этого месяца – *Summoning Wars*, блестящая ролевая игра в жанре экшн. Она использует боевую систему наведи-и-щелкни, основанную на четырех видах персонажей. Есть здесь также многопользовательский режим для вас и семерых соратников.

Ищете чего-то полегче? Тогда – в небеса, к *Airrace*. Это тренажер полетов, и ваша задача – гонки в воздухе среди ориентирных вышек. Особый шарм ему придает смесь серьезности и удовольствия: полет смоделирован весьма реалистично, но не нужно нажимать на сотни кнопок, чтобы просто заставить самолет хоть что-то сделать.

Для поклонников головоломок у нас есть *NullpoMino*, клон *Tetris* с широкими возможностями настройки, включающий множество разнообразных игровых режимов. Еще у нас есть *Peces*, геометрическая головоломка, где создаются фигуры из набора многоугольников.

Если вам надо чем-то занять детей, обратите внимание на *GCompris*, пакет обучающих и развивающих программ. Он многоцветный, занимательный, и включает более 100 игр.



› Эпичный провал попытки обойти вышку в *Airrace*.

На диске

Ее пришлось-таки подождать, но вот она — новая версия Debian!

Дистрибутив Linux

Debian 6

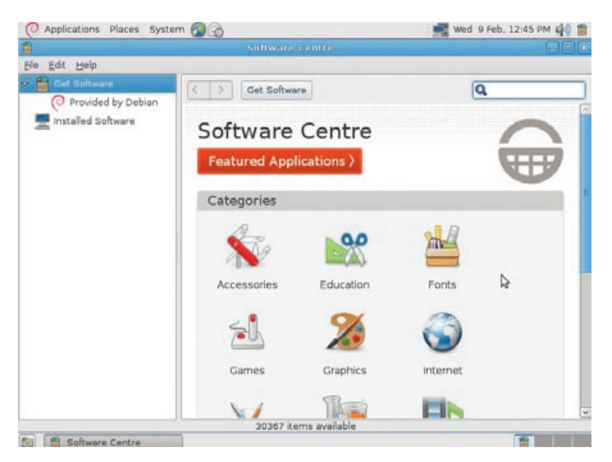
В мире Linux ходит шутка «для внутреннего употребления», что релизы Debian встречаются реже, чем куриные зубы – и правда, Debian 5 мы включали в состав диска аж в **LXF118** (май 2009). Однако всему сообществу известно, что тут есть и плюс: Debian невероятно надежен и хорошо тестирован. Месяцы исправления ошибок и отладки приходятся на последние стадии цикла релиза Debian, и в итоге получается дистрибутив, столь известный своей стабильностью, что клоны Ubuntu принимают его за основу.

Это значит, что Debian не включает самый свежий передовой код, поскольку пакеты должны пройти процесс весьма требовательной проверки качества. Зато релизы поддерживаются несколько лет, и поэтому Debian – популярный выбор для серверов и предприятий. Вы узнаете больше о Debian 6 (он же Squeeze, в честь персонажа из «Историй игрушек»), прочтя наш обзор. А вот как заставить его работать.

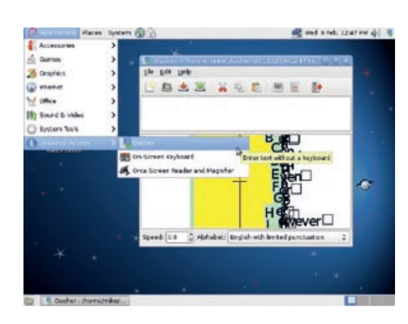
Debian 6 загружается прямо с **LXFDVD** в режиме live, и вы сможете опробовать его без установки на жесткий диск. (Если ваш ПК не загружается с DVD, а сразу переходит на жесткий диск, загляните в раздел Help/New to Linux на DVD: там есть инструкции по изменению порядка загрузки в BIOS.) Когда загрузится рабочий стол, можете начинать знакомство с имеющимися в нем программами и установить Debian на жесткий диск, следуя приведенным ниже инструкциям.

Про разбиение диска на разделы: как и любой операционной системе, Debian нужно место на жестком диске. Если вы отводите всю машину целиком под Linux, можете просто указать программе установки занять весь диск, что очень мило и просто. Если вы пока работаете в Windows, выделите место для Linux на стадии разбиения диска на разделы в процессе установки. В конечном итоге вы получите систему с двойной загрузкой и будете выбирать Linux или Windows при загрузке компьютера. И хотя эта процедура практически всегда проходит без неприятностей, мы все же рекомендуем сделать резервные копии важных данных в Windows перед началом установки – на случай цунами!

Debian использует интернет-репозитории пакетов для обновлений и добавления ПО. После установки перейдите в System > Administration > Software Sources и проверьте, включены ли нужные репозитории. Выберите ближайший к вам сервер из списка Download From, затем проверьте официально поддерживаемые и DFSG-программы. Также перейдите на вкладку Updates и проверьте обновления безопасности. По нажатии на Close Debian быстро просканирует репозитории в Интернет, и вы получите доступ к новейшим программам.



› Отношения Debian–Ubuntu строятся в обе стороны, как показывает включение Canonical Software Centre.



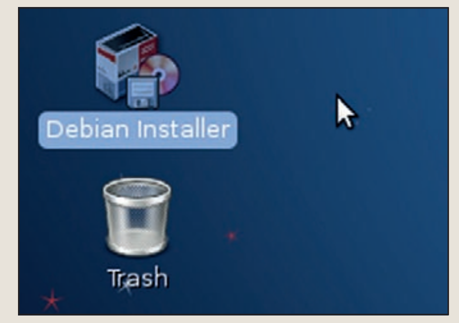
› Будучи «универсальной ОС», Debian всегда заботился о доступности для пользователей с ограниченными физическими возможностями.

Шаг за шагом: Устанавливаем Debian 6



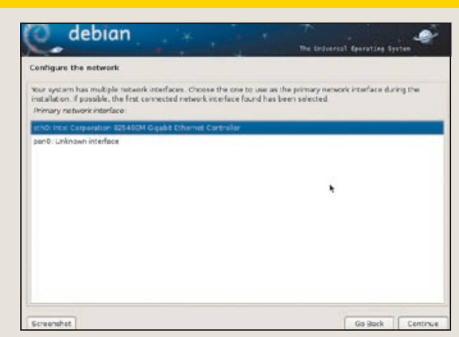
1 Загрузка

Загрузите свой ПК с **LXFDVD** и нажмите на Enter в пункте Debian Live, увидев это загрузочное меню. Если возникнут проблемы, перезагрузите компьютер и выберите опцию failsafe.



2 Рабочий стол

После загрузки дистрибутива перед вами появится рабочий стол Gnome, где вы сможете познакомиться с имеющимися программами. Чтобы запустить установку, дважды щелкните по значку Debian Installer.



3 Сеть

Выберите язык и раскладку клавиатуры. После этого вы перейдете к сетевым настройкам, где можно с успехом нажать на Continue и принять настройки по умолчанию.

Как бы мне?..

Если вы – абсолютный новичок в Linux, у вас могут возникнуть сомнения по поводу того, как справиться с определенными задачами после установки Debian. Вот список наиболее обычных задач и способов их решения…

» **Работа в сети** Щелкните по Applications > Internet > Iceweasel, чтобы запустить версию *Firefox*, второго по популярности в мире браузера, поддерживаемого сотнями расширений.

» **Работа с файлами** Перейдите в Places > Home Folder для хранения и доступа к вашим личным документам.

» **Воспроизведение музыки и видео** Попробуйте *Movie Player* в меню Applications > Sound & Video в качестве простого видеопроигрывателя, или *Rhythmbox* в качестве всестороннего музыкального менеджера.

» **Редактирование фото** Загляните в Applications > Graphics > Shotwell Photo Manager: это отличный инструмент, позволяющий навести порядок и отредактировать ваши цифровые фотографии.

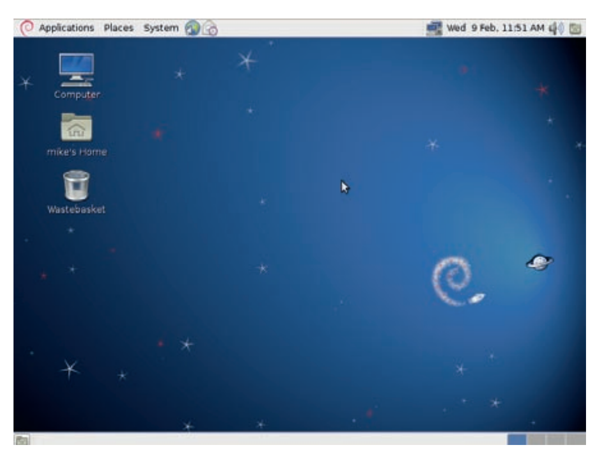
» **Чат онлайн** В меню Internet попробуйте *Empathy*. Это очень богатый функциями клиент для обмена быстрыми сообщениями, который поддерживает все популярные протоколы (MSN, AIM, Google и т.д.).

» **Настройка системы** В меню System вы найдете подменю Preferences и Administration с инструментами для настройки сети, управлению пользователями и многого другого.

» **Установка новых программ** Нажмите на System > Administration > Software Centre, чтобы просмотреть, какие программы имеются в Интернет. Если вам нужен более продвинутый менеджер пакетов, попробуйте *Synaptic*.

» **Выход и выключение** Щелкните по меню System, где имеются опции выхода из текущего пользовательского аккаунта или выключения компьютера.

Debian раньше пользовался репутацией дистрибутива, сложного для начинающих пользователей Linux, но это перестало быть проблемой. В меню System имеется множество графических инструментов настройки, а онлайн – огромное количество подсказок. На www.debian.org/doc есть очень подробное руководство по установке и справочник по операционной системе – это одни из лучших материалов в мире Linux. При наличии проблем попробуйте задать вопрос на наших дружелюбных форумах на www.linuxformat.com/forums. Подписка бесплатна, и мы не занимаемся рассылкой спама. Подробно опишите проблему, и другие пользователи Debian обязательно укажут вам верный путь. Удачи! LXF



› **Нам нравится новая «космическая» тема; надеемся, она не скроет от людей, что Debian — очень серьезный дистрибутив.**

Не пропустите...

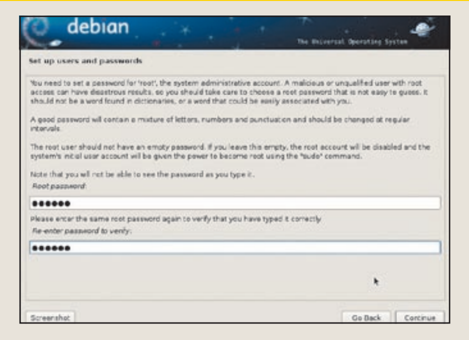


OpenOffice.org 3.2

Ура! Самый свежий релиз Debian застрял на *OOo 2*, так что это — желанное обновление для пользователей настольных ПК.

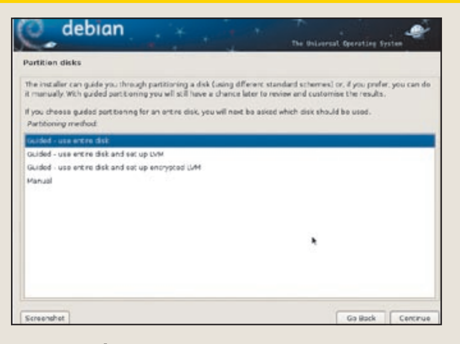
Chromium

Теперь можно установить быстрый как молния браузер от Google из *Software Centre* — просто запустите поиск «chromium-browser».



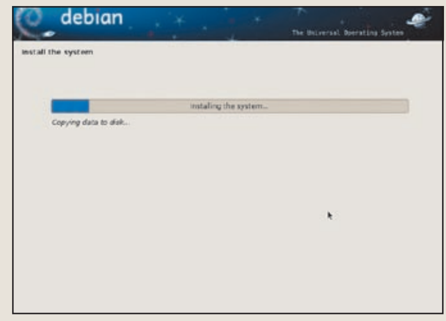
4 Учетные записи

Далее вам предложат задать пароль для пользователя root (администратора) и создать учетную запись обычного пользователя. Помните, что регистр важен!



5 Разбиение на разделы

Экран разбиения диска на разделы предложит вам различные опции в зависимости от раскладки вашего диска. При большом опыте разбиения диска на разделы вы всегда можете выбрать Manual (Вручную).



6 Копирование

Когда файлы будут скопированы, выберите опцию Install Grub. Когда эта стадия завершится, щелкните по System > Shut Down, чтобы перезагрузить свой компьютер, и затем запустите Debian с жесткого диска.



Охранные системы на базе GNU/Linux

Компания «Нордавинд» существует с 2004 года и является пионером в области разработки и производства компонентов для охранных систем на базе высоконадежной операционной системы GNU/Linux

Наши услуги и продукты

- Проведение НИР
- Разработка и изготовление систем охранного телевидения и сопутствующих компонентов
- Проектирование сложных систем и комплексов
- Широкий ассортимент готовой продукции

Наши системы видеонаблюдения предназначены для:

Паркингов
Зданий
Бизнес-центров
Торговых комплексов и складов
Открытых площадок
Жилых домов
Объектов повышенной секретности
Режимных предприятий

Сертификаты **ФСТЭК** и **ФСБ**

4 канальная система видеонаблюдения стоимостью **40 тыс. руб.** — **в 2 раза дешевле**, чем предыдущие технологии.



ТелеВизард
Многофункциональный сервер общего назначения.



ТелеВизард Авто
Самая бюджетная и высокоэффективная система распознавания автомобильных номеров.



Источники видеосигнала
Комплекс eyeSense, в состав которого входят видеокамера и объектив. Дополнительное оборудование — ИК-прожектор и климат-защита.

«Нордавинд Северо-Запад»
Научно-производственная компания

Тел. +7 (812) **309 0686**
Тел./факс +7 (812) **640 4990**

http://**spb.nordavind.ru**
e-mail: **spb@nordavind.ru**

Приглашаем к партнерству дилеров и проектировщиков систем

Информация о диске

Внимательно прочтите это перед тем, как использовать LXFDVD!

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

- » **имя_программы-1.0.1.i386.rpm** – вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- » **имя_программы-1.0.1.i386.deb** – такой же пакет, но уже для Debian;
- » **имя_программы-1.0.1.tar.gz** – обычно это исходный код;
- » **имя_программы-1.0.1.tgz** – тот же файл, что и выше по списку: «tgz» – это сокращение от «tar.gz»;
- » **имя_программы-1.0.1.tar.bz2** – тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя_программы-1.0.1.src.rpm** – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя_программы-1.0.1.i386.FC4.RPM** – двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя_программы-1.0.1.ppc.Suse9.rpm** – двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя_программы-devel-1.0.1.i386.rpm** – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru





» Содержание

LINUX FORMAT

ДИСТРИБУТИВЫ
Debian 6 Новый релиз известнейшего дистрибутива
INX 1.1 «Учебный» дистрибутив для работы в командной строке
KDE 4.6 Live Загрузочный диск с самой новой версией KDE
openSUSE 11.4 Новая версия популярного дистрибутива (для x86-64)

РАБОЧИЙ СТОЛ
4Pane 0.8.0 Файловый менеджер
Back In Time 1.0.6 Утилита резервного копирования
CLFSWM 1102 Менеджер окон, позволяющий работать без мыши
ExifTool 8.48 Средство для работы с метаданными изображений
KDE 4.6.0 Исходные тексты новой версии
LibreOffice 3.3.0 Офисный пакет, ветка OpenOffice
MuseScore 1.0 Редактор музыки
PCManFM 0.9.9 Файловый менеджер
PieDock 1.1.0 Панель задач, отображаемая вокруг курсора
Rox-Filer 2.10 Файловый менеджер
Xfce 4.8 Исходные коды новой версии

РАЗРАБОТКА
Box 0.2.1 Язык программирования векторной графики
Dwdiff 1.9 Надстройка над diff, работающая на уровне слов, а не строк
GTK 2.24.0 Набор средств программирования графических интерфейсов
KDevelop 4.2.0 Среда разработки для KDE
Mokoi 0.90 Кроссплатформенная среда разработки игр
PyChecker 0.8.19 Средство отладки для Python
Submin 2.0 Web-интерфейс к репозиториям Subversion

ИГРЫ
Airrace 0.2 Гонки в воздухе
GCompris 9.5 Обучающая игра для детей от 2 до 10 лет
NullpoMino 7.5.0 Головоломка в стиле Tetris
Peces 4.0.2 Традиционная китайская игра Tangram
Summoning Wars 0.54 Многопользовательская ролевая игра

HOTPICKS
Bandwidth 0.26b Средство тестирования производительности памяти
Din 1.4.2 Синтезатор звуков и музыки
DMIdecode 2.11 Средство получения информации об аппаратуре
Figlet 2.2.4 Средство печати ASCII-графики
Hotot 0.9.5 Клиент микроблоггинга
Kernel Panic 4.1 Игра в стиле Doom на «околокомпьютерную» тему
KiCad snapshot Средство проектирования электроники
PokerTH 0.8.3 Игра, многопользовательский покер по Интернет
Powertop 1.97 Средство мониторинга энергопотребления
Smartie-utils 1.0.0 Утилиты для работы с дисплеем LCD Smartie
Wings3D 1.4.1 Средство 3D-моделирования

INTERNET
DansGuardian 2.10.1.1 Фильтр web-контента
Linphone 3.4.0 Видеотелефон с поддержкой SIP
Rspamd 0.3.6 Система блокировки спама
Sambascan 0.5.0 Средство поиска SMB-серверов в сети
Todoyu 2.0 Система управления проектами

СИСТЕМА
FSlint 2.42 Средство очистки и оптимизации файловой системы
HDT 0.4.1 Средство для получения информации об аппаратуре
Malmon 0.3 Система обнаружения эксплойтов в реальном времени
Partimage 0.6.9 Средство для работы с образами дисковых разделов
Webmin 1.530 Web-интерфейс для администрирования Linux
ZUPS 1.2 Средство мониторинга источников бесперебойного питания

Пожалуйста, перед использованием данного диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИИ? Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕФЕКТНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данном диске, обращайтесь, пожалуйста, по адресу disks@linuxformat.ru

Тираж изготовлен ООО «Уральский электронный завод». 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПТР ВАФ 77-15.

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – метка; затем после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root и выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере с Windows, Mac OS X, AmigaOS или любой другой ОС.

Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было бы записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков: подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт разработчика дистрибутива.



Пропустили номер?

» Мир свободного ПО богат и разнообразен, а потому далеко не все можно вместить в рамки одной статьи. Linux Format обходит эту проблему, публикуя серии статей по самым актуальным вопросам. Но что делать, если вы поймали интересующий вас материал на середине? Обратитесь в «Линуксцентр» по адресу www.linuxcenter.ru и закажите желаемый номер журнала! Он доставляется как в печатной, так и в электронной форме, поэтому с момента открытия браузера и до получения нужного вам выпуска **LXF** может пройти не более нескольких минут!

Прямо сейчас для заказа доступны следующие номера:



LXF139/140
Январь 2011

250 руб.

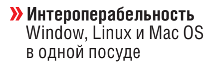
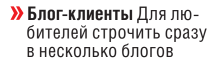
- **Linux и безопасность** Как позаботиться о защите вашей системы
- **Дистрибутивы-экзоты** Обзор специализированных дистрибутивов Linux
- **Xfce** Легкий и быстрый менеджер окон, не всем заметный в тени Gnome и KDE
- **8 «нет» о GPL** Заблуждения насчет популярной лицензии опровергает российское законодательство

LXFDVD: Ubuntu 10.10 (DVD с тройной загрузкой: LXF Ubuntu, Xubuntu, Kubuntu), Fedora 14, OpenBSD 4.8

Печатная версия: shop.linuxformat.ru/lxf_139_140/
PDF-версия: shop.linuxformat.ru/elxf_139_140/

LXF141
Февраль 2011

250 руб.

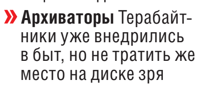
- **Интероперабельность** Window, Linux и Mac OS в одной посуде
- **Блог-клиенты** Для любителей строчить сразу в несколько блогов
- **Wayland** Новый сервер графики Linux, который идет на смену X
- **Drupal 7** Как построить сайт с легкостью необыкновенной

LXFDVD: Sabayon, CentOS, Tiny Core, Myth TV, игры и прочее

Печатная версия: shop.linuxformat.ru/lxf_141/
PDF-версия: shop.linuxformat.ru/elxf_141/

LXF142
Март 2011

250 руб.

- **Резервное копирование** Как уберечь ваши драгоценные данные
- **Архиваторы** Терабайтники уже внедрились в быт, но не тратить же место на диске зря
- **Компьютерная верстка** В Linux это постепенно становится реальностью
- **Тренинги Linux** Майк готовит публику к сдаче экзамена на сертификат

LXFDVD: Puppy Linux, SystemRescueCd, Lubuntu, Clonezilla, VirtualBox и прочее

Печатная версия: shop.linuxformat.ru/lxf_142/
PDF-версия: shop.linuxformat.ru/elxf_142/

Ну, а если вы хотите быть уверенными, что не пропустите ни один номер журнала — оформите подписку! Помните, что все подписавшиеся на печатную версию журнала через www.linuxcenter.ru или shop.linuxformat.ru получают электронную версию в подарок!

Подписывайтесь на журнал на www.linuxformat.ru/subscribe/

Телефоны отдела подписки: Санкт-Петербург (812) 309-06-86, Москва (499) 271-49-54

Специальное предложение

Купите подборку журнала!

К нам в редакцию периодически приходят письма с вопросами, где можно купить предыдущие выпуски **LXF**. Если вы тоже озадачены этим вопросом, то в интернет-магазине «ГНУ/Линуксцентра» продолжается продажа журналов за 2010 год. Вы можете приобрести как отдельные номера изданий, так и подписки на 6 или 12 месяцев.
Спешите — журналов осталось не так уж много!

shop.linuxformat.ru





Олимпийский ЕГЭ

Не повлияет ли использование Linux и свободного ПО на итоговую аттестацию учащихся и участие в олимпиадном движении? Разбирается **Александр Казанцев**.



Наш эксперт

Александр Казанцев
К. т. н., доцент кафедры информатики Глазовского государственного пединститута, руководитель проекта EduMandriva, автор локализаций (и разработок) для Mandriva, *Klavaro*, *Gambas2* и *LXDE*.

Внедрение свободного ПО в целом и Linux в частности постоянно сталкивается с противодействием учителей и органов образования. Вторая по популярности отговорка после «это не нужно работодателям» – дети не сдадут ЕГЭ и не смогут участвовать в олимпиадах.

На самом деле это именно отговорка – мы постараемся разобрать задачи из единого государственного экзамена и посмотреть, что предлагается учащимся к использованию на российских и международных олимпиадах, и сделаем вывод: правы ли те, кто вставляет внедрению палки в колеса.

ЕГЭ и ОС

Первым делом рассмотрим головную боль учителей старших классов – подготовку учащихся к сдаче единого государственного экзамена по информатике и ИКТ. Влияет ли использование Linux без привычных *Word*, *Excel* и Delphy на подготовку?

Для этого зайдем на сайт www.ege.edu.ru и скачаем демонстрационный вариант. Что же мы имеем? Часть А содержит вопросы по файловой системе, но оперирует стандартными понятиями файла и расширения, а также масками, работающими на всех современных файловых системах. Windows, конечно же, встретился, но в контексте забывчивой девочки, которая при работе с Linux не потеряла бы свой пароль. Программные задания даны на четырех языках – Бейсике, Паскале, Си и алгоритмическом, которые все имеются в свободном ОС. И даже электронные таблицы и базы данных представлены как просто электронные таблицы или базы из таблиц, без привязки к конкретной программе. Вопросы по Интернет-технологиям по их принципу будут являться кросс-платформенными. Исполнитель представлен Роботом, который является аналогом Черепашки и Робота из КУМИРа, и все это ПО присутствует в Linux.

Часть B также не содержит зависимых от ОС заданий. Часть C содержит только задачи на алгоритмы, и по правилам проведения ЕГЭ может быть решена на любом языке (хоть на PHP, Python или даже Perl).

Как видно, ничего, способного помешать ученикам, изучающим информатику и ИКТ только на базе свободного ПО, просто не существует. Считаем, что этот миф развеян.

Олимпиадное движение

Олимпиадное движение в нашей «стране непуганых пиратов» до последнего момента было направлено на использование того же программного обеспечения (в большей части нелицензионного), что и в школьной программе. Это было удобно преподавателям и организаторам, но привело к катастрофическим результатам. Международное олимпиадное движение в то же время двигалось в сторону свободного ПО и использования Linux и языков программирования, отличных от Turbo Pascal и Visual Basic. В последние годы ситуация стала выправляться, и требования ко внутренним олимпиадам стали более приближены к международным.

> «Требования к олимпиадам приближены к международным.»

Рассмотрим требования к используемым программным средам. В методических рекомендациях на этот год четко прописано, что на региональных и российских олимпиадах рекомендуется использовать следующие виды ПО: Borland Pascal, Borland C++, Microsoft Visual C/C++; Microsoft Visual Basic, FreePascal; GNU C/C++, Java, Eclipse, Code::Blocks. Как видно, список включает множество сред и языков, которые есть в Linux и даже работают лучше в Linux.



Архив LXF за 2010 год! стр. 107

Чему готовить участника?

Участник олимпиады с прицелом на международное участие должен:

- Владеть технологией программирования на языке С.
- Владеть математическими основами для решения алгоритмических задач по темам, обозначенным содержанием программы подготовки к олимпиадам по информатике.
- Свободно владеть набором вслепую в отладчике среды С.
- Знать принципы написания тестов к программам. Тренировать дома работу в системе отладки задач.
- Быть знакомым со средой Linux, особенностями отладки в этой среде по сравнению с MS.

Требования к задачам – наличие ввода (с потока ввода или текстового файла и вывода в текстовый файл или поток вывода) – все кросплатформенно. Если задача имеет стандартную алгоритмику (то есть не затрагивает принципов ООП, элементов GUI и API или графики), то разницы в решениях на Linux и Windows не должно быть. Разница в программировании (именно используя связку «стандартный ввод – решение – стандартный вывод») заключается только в разном формировании путей файловой системы и возможных проблемах с локалью.

При определении состава программного обеспечения для регионального этапа олимпиады региональным предметно-методическим комиссиям и жюри рекомендуется учитывать тот факт, что одной из основных целей всероссийской олимпиады школьников является отбор наиболее талантливых обучающихся для участия в составе сборной команды страны на международной олимпиаде по информатике. В этой связи организаторам регионального и российского этапов олимпиад необходимо предоставить лучшим школьникам региона возможность проявить себя в рамках сред программирования, используемых на международной олимпиаде по информатике (см. врезку *Чему готовить участника*?).

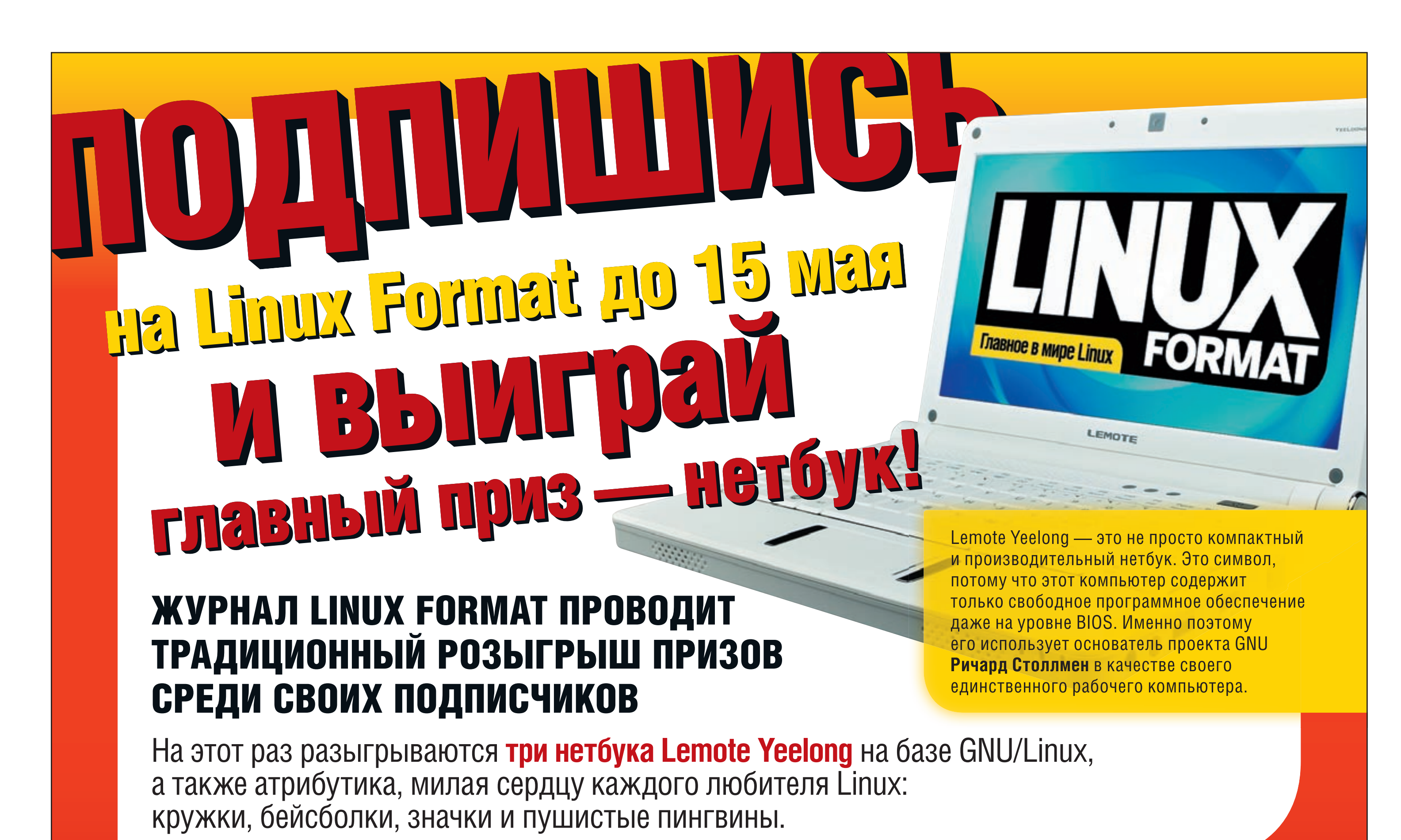
Если мы посмотрим рекомендуемый список ПО международной олимпиады, то обнаружим следующее:

- Ubuntu 10.4 i386; GNOME.
- Web-браузер: *Firefox*.
- Редакторы: *mcedit*, *joe*, *vim*, *kate*, *kwrite*, *emacs*, *lazarus*, *gedit*, *nano*, *scite*, *codeblocks*, *geany*, *fp*.
- Компиляторы: *gcc 4.4.3*, *g++ 4.4.3*, *fpc 2.4.0*.
- Отладчики-дебаггеры: *gdb*, *ddd*, *valgrind*.

Как видно, и ЕГЭ, и олимпиады не должны вызывать проблем при использовании свободного ПО. Хотя в списке олимпиадных сред программирования еще много проприетарных решений, международное олимпиадное движение их не использует, и мы, если хотим получить хорошие результаты, должны следовать мировым тенденциям. LXF

Где взять подробности?

Информацию «из первых рук» можно получить в лектории Владимира Михайловича Кирюхина – члена Центрального оргкомитета Всероссийской олимпиады школьников Минобрнауки России, председателя Центральной предметно-методической комиссии по информатике, по адресу http://metodist.lbz.ru/lections/6/.



Вторая родина – Россия



Linux нравится разным пользователям, в том числе и самым маленьким. О новом дистрибутиве для детей рассказывают **Сергей Комков** и **Наталья Семенова**.



Наш эксперт

Наталья Семенова
Кандидат педагогических наук. Ее научные интересы – исследовательская деятельность учащихся и методики начального образования.



Наш эксперт

Сергей Комков
ИТ-профессионал со стажем более двадцати лет. Считает, что свободное ПО, особенно ОС для детей, должно внедряться методично.

В начале сентября 2010 в Рунете появились первые сообщения о новой операционке для детей – DoudouLinux. Сотрудники Института прикладной информатики Томского государственного педуниверситета (ТГПУ) заинтересовались проектом. После всестороннего изучения было принято решение перевести систему на русский язык. В конце октября завершился первый этап локализации, в рамках которого был переведен интерфейс этой системы и основных служебных утилит, а также создано русское зеркало официального сайта проекта – doudoulinux.org.

Почему ДудуЛинукс показался нам лучшей ОС для малышей?

- Максимальная простота (можно не знать буквы, никаких двойных щелчков и операций с файлами, нет выпадающих меню).
- Направленность на обучение самых маленьких детей владению компьютером, чтобы они поняли связь между мышью/клавиатурой и курсором на экране.
- Нет бесполезных или перегруженных приложений, только 50 программ, соответствующих потребностям ребенка.
- В последних версиях присутствует родительский контроль, а это большая редкость для детских ОС.
- DoudouLinux базируется на Linux Debian, которая дает стабильность и обеспечивает долговечность проекта.
- DoudouLinux направлен на то, чтобы сделать компьютер доступным для всех детей с самого раннего возраста (два года).
- Минимальные требования к оборудованию: память 256 МБ, процессор 800 ГГц, разрешение монитора 800×600.

Систему можно записать на USB-носитель (или компакт-диск) и загружаться с него, не касаясь ни данных, хранящихся на компьютере, ни установленной операционной системы. Так что можно носить его с собой, спокойно давая детям играть на любом компьютере. DoudouLinux можно одолжить, подарить, скопировать по своему желанию – совсем как на детской площадке! Это абсолютно законно, потому что DoudouLinux может свободно копироваться и распространяться на условиях общедоступной лицензии (GPL). Руководителем проекта является французский энтузиаст Жан-Мишель Филипп.

Программы ДудуЛинукс тематически делятся на 4 группы:

- Обучающие (в частности, *Gamine*, *Pysycache*, *Childsplay*, *gCompris*, *TuxPaint* и другие)
- Игры (обучающие, развлекательные, познавательные)
- Мультимедиа (клавиатура, ударная установка, проигрыватель)
- Для работы (словарь, текстовый редактор, браузер и другие)

Главный экран системы предлагает на выбор семь сессий – пять из них запускают по одному-единственному приложению (*Gamine*, *Psysycache*, *Childsplay*, *gCompris*, *TuxPaint*), а две представляют собой наборы приложений – минимальный и полный. В этих сессиях приложения организованы в тематические вкладки.

В ноябре началась российская жизнь ДудуЛинукс. В ТГПУ прошла презентация системы, вызвавшая широкий резонанс среди томских мам и пап. Волну интереса к ДудуЛинукс вызвали телеви-



› **Официальный сайт проекта. Один из семи языков — русский.**

зионный ролик Вести-Томск и публикация на сайте mama.ru и новостной ленте агентства Интерфакс.

Тогда же начался второй этап локализации – перевод программ, входящих в дистрибутив, а также звуковых сообщений. Кроме локализации, в ИПИ начали планомерное тестирование и продвижение системы. В результате осенью 2010 года резко выросло количество скачиваний DoudouLinux; к февралю 2011 система была скачана более 30 000 раз, из них треть – на русском языке.

Далее, появились намерения осмыслить состав сессий с точки зрения их дидактической ценности. Коллеги с кафедры теории и методики начального образования проанализировали программы и предложили дополнить систему наборами обучающих сессий для школьников и дошкольников.

ДудуЛинукс позволяет осуществить деятельностный подход в развитии ребенка, а также обеспечивает индивидуальный темп развития.

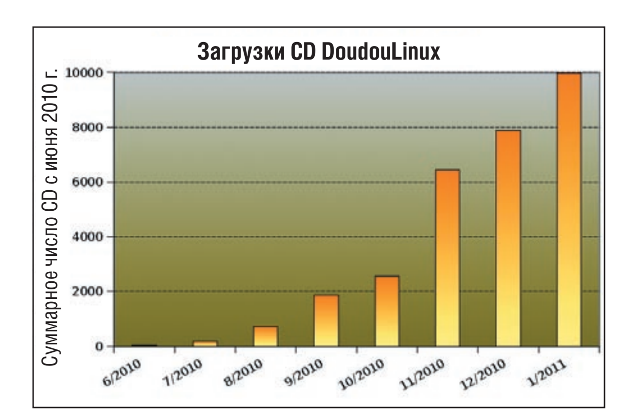
За основу обучающих сессий были взяты известные наборы игр *gCompris* и *Childsplay*. Из них мы отобрали игры, обладающие наибольшей дидактической ценностью. Затем разбили их по трем категориям:

1. Освоение компьютера (*Gamine*, *Pysycache* и др.)
2. Обучение и развитие мышления (разделы Математика и логика, Окружающий мир, Обучение грамоте, Эстетическое развитие)
3. Игры и развлечения

Программы первой категории позволяют формировать у дошкольников умение работать с мышью, клавиатурой, ориентироваться на рабочем столе, а детям более старшего возраста – совершенствовать эти умения. Сюда относятся такие приложения, как *Gamine*, *Pysycache* и др.

Программы второй категории включают задания, направленные на развитие зрительного и слухового восприятия, зрительной и слуховой памяти, на развитие произвольного внимания, а также творческого и логического мышления и воображения. Каждый родитель и педагог стремится к развитию этих познавательных процессов, и как одно из средств такого развития Дудулинукс достаточно эффективен. Кроме того, к данной категории можно отнести ряд приложений, направленных на первичное ознакомление детей с новыми для них знаниями – об окружающем мире, о цифрах и счете, буквах. Ребенок включается в интересную для него игровую деятельность и в процессе такой игры усваивает материал, который впоследствии станет прочной основой для дальнейших знаний, обучения и развития. К таким программам относятся, в частности, *TuxPaint* и *kGeography*.

Более опытных детей и школьников можно привлечь заданиями иного характера, требующими уже сформированных умений и некоторых знаний. Здесь дети могут найти им применение, что позволит не только закрепить данный материал, но и сформирует положительную самооценку у ребенка – позицию «я могу», «я успешен», что немаловажно для его дальнейшего саморазвития и совершенствования.



› **Популярность дистрибутива возрастает достаточно быстро.**



› **DoudouLinux предлагает набор образовательных игр, простых в использовании, чтобы сделать обучение веселым.**

Третья категория также значима, поскольку такие программы являются своего рода «переменкой», позволяя сменить вид деятельности, включиться в более азартную игру, отработать юным пользователям навыки работы с мышью и клавиатурой, которые они приобрели, действуя в рамках специальных приложений – например, Мистер Клубень.

Таким образом, ДудуЛинукс позволяет педагогам и родителям успешно реализовывать основные задачи – развитие и обучение, формирование навыков работы с компьютером и организация досуга своего ребенка. А ребенок может самостоятельно осваивать данные приложения и свободно выбирать то задание, которое для него наиболее интересно в данный момент, без страха что-то нарушить в работе компьютера. Это позволяет максимально индивидуализировать процесс развития каждого ребенка.

Планируется экспериментальное использование обучающих сессий в учебном процессе в МОУ «Камскополянская средняя общеобразовательная школа № 2 с углубленным изучением отдельных предметов» (Татарстан).

Наконец, в ИПИ задумались о переводе системы на другие языки. В рамках языковой практики студенты ТГПУ и ТПУ активно работают над немецкой версией DoudouLinux, а также сайта проекта.

На днях увидит свет новая версия ДудуЛинукс – 2011-02. Теперь система поддерживает 8 языков: английский, арабский, испанский, китайский, сербский, румынский, русский, французский.

Вот чем она отличается от предыдущей.

- Русифицирован весь интерфейс системы и служебных утилит.
- Русифицированы обучалки для малышей – *Gamine*, *Pysycache*, *kTuberling*, *kLettres* (в том числе звуки).
- Разработан мастер настройки списка сессий стартового экрана DoudouLinux.
- Устранены недочеты в Справке по DoudouLinux.
- Исправлены ошибки в системных скриптах и настройках приложений. LXF



Томские инициативы



Институт прикладной информатики (ИПИ) (заместителем директора по развитию которого является Сергей Комков) при Томском государственном педагогическом университете (Наталья Семенова – сотрудник кафедры педагогики и методики начального образования Педагогического факультета университета) был организован в 2001 году. ИПИ – официальный партнер компании AltLinux и активно участвует в проектах по внедрению СПО в учреждениях образования. В частности, это федеральные проекты «Разработка и апробация в пилотных субъектах РФ пакета СПО для использования в общеобразовательных учреждениях РФ в 2007–2008 годах» и «Обучение специалистов сферы образования внедрению пакета СПО и использованию СПО в учебном процессе», а также региональный проект «Внедрение СПО в образовательных учреждениях Томской области» на педагогическом факультете ТГПУ.



В майском номере

MythTV: Ящик под контролем

Загрузите телевизор и смотрите что угодно и когда угодно. Звезды сериалов сроду не выглядели так клево!

Обращение в Linux-веру

Как найти друзей и убедить людей использовать Linux. А для особо смелых – даже FreeBSD.

Сравнение: графопостроители

Заточите карандаши, нанесите оси: грядет пиршество графических ухищрений.

Содержание последующих выпусков может меняться без уведомления.



Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор
Кирилл Степанов info@linuxformat.ru
Литературный редактор
Елена Толстякова
Переводчики
Илья Аввакумов, Ольга Кокорева, Светлана Кривошеина, Алексей Опарин, Валентин Развозжаев, Елена Толстякова, Татьяна Цыганова
Редактор диска
Кирилл Степанов
Верстка, допечатная подготовка
Сергей Рогожников
Технический директор
Денис Филиппов
Директор по рекламе
Владимир Савельев advert@linuxformat.ru
Генеральный директор
Павел Фролов
Учредители
Частные лица
Издатели
Виктор Федосеев, Павел Фролов

Отпечатано в ООО «Скай ЛТД»
196210, Санкт-Петербург, Взлетная ул., 11, корп. 2, лит. А
Тел.: (812) 677-98-03
Заказ 3598

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Пол Хадсон [Paul Hudson] paul.hudson@futurenet.com
Редактор обзоров Грэм Моррисон [Graham Morrison] graham.morrison@futurenet.com
Редактор диска Майк Сондерс [Mike Saunders] mike.saunders@futurenet.com
Художественный редактор Эфраин Эрнандес-Мендоса [Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com
Литературный редактор Эндрю Грегори [Andrew Gregory] agregory@futurenet.com
Подготовка материалов Джоно Бэкон [Jono Bacon], Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Энди Ченнел [Andy Channelle], Марко Фиоретти [Marco Fioretti], Эндрю Грегори [Andrew Gregory], Грэм Моррисон [Graham Morrison], Боб Мосс [Bob Moss], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Майк Сондерс [Mike Saunders], Ник Вейч [Nick Veitch], Грэм Уэлдон [Graham Weldon], Евгений Балдин, Андрей Боровский, Александр Казанцев, Сергей Комков, Андрей Кузьменко, Антон Резник, Наталья Семенова, Андрей Ушаков, Алексей Федорчук
Художественные ассистенты Стейси Блэк [Stacey Black], Ник Кокс [Nick Cox], Фил Хейкрафт [Phil Haycraft]
Иллюстрации Крис Хедли [Cris Hedley], Крис Винн [Chris Winn], Elly Walton Illustrations, iStockPhoto

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel +44 01225 442244 Email: linuxformat@futurenet.com
РОССИЯ:
Санкт-Петербург (редакция):
Лиговский пр., 50, корп. 15
Тел. +7 (812) 309-06-86
Представительство в Москве:
Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)
Тел./факс +7 (499) 271-49-54
По вопросам сотрудничества, партнерства, оптовых закупок:
partner@linuxcenter.ru

Новое поколение средств защиты

Межсетевые экраны ССПТ, не имеющие IP-адреса

ССПТ-2 — это сертифицированное ФСТЭК И ФСБ средство защиты информации нового поколения, реализующее функции межсетевого экрана, но при этом остающееся «невидимым» для любых протоколов и тестовых воздействий, что достигается за счет отсутствия физических и логических адресов на его фильтрующих интерфейсах. **ССПТ-2 невозможно обнаружить никакими известными средствами удаленного мониторинга сети.**

Скрытность функционирования межсетевого экрана повышает надежность системы защиты в целом и существенно упрощает процедуру установки ССПТ-2 в компьютерные сети и функционирующие на их основе информационные и телематические системы.

Защита для высокоскоростных корпоративных сетей Ethernet 100/1000 Мбит/с

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



Назначение устройства

Основное средство защиты для реализации различных политик информационной безопасности с помощью:

— фильтрации пакетов на канальном, сетевом, транспортном и прикладном уровнях;

— управления транспортными соединениями между отдельными узлами ЛВС или виртуальной ЛВС (VLAN);

— контроля контента данных на прикладном уровне с учетом направления, времени и типа протоколов передачи трафика.

Дополнительное устройство защиты для:

— обеспечения безопасности функционирования ранее установленных в компьютерной сети средств защиты и устройств маршрутизации;

— мониторинга трафика с возможностью анализа данных регистрации пакетов по различным критериям и интеграции с IDS;

— обеспечения функционирования сетевых распределенных телематических приложений и GRID-ресурсов.





Москва +7 (499) 271-49-54

Санкт-Петербург +7 (812) 309-06-86

Linux-эксперт для вашего бизнеса. www.linuxcenter.ru

